

**HP 2392A**

**Display Terminal**

**HP 2394A**

**Data Entry**

**Terminal**

**Reference Manual**

## **RADIO FREQUENCY INTERFERENCE (RFI) STATEMENTS**

### **FOR THE UNITED STATES**

#### **Federal Communications Commission (FCC)**

The US Federal Communications Commission (in 47 CFR 15.818) has specified that the following notice be brought to the attention of the users of this product.

<b>WARNING</b>
----------------

This equipment generates, uses, and can radiate radio frequency energy and if not installed and used in accordance with the instruction manual, may cause interference to radio communications. It has been tested and found to comply with the limits of a Class A computing device pursuant to Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference when operated in a commercial environment. Operation of this equipment in a residential area is likely to cause interference, in which case the user at his own expense will be required to take whatever steps may be required to correct the interference.

### **FOR EUROPE**

#### **Verband Deutscher Elektroniker (VDE)**

This equipment has been certified to comply with the requirements of DIN5 7u871/VDE 0871/6.78 level B of German PTT regulations FTZ III 5/82, and carries the Radio Protection Mark.

## Publications History

Revision 0 (original) ..... April 1985

This manual applies to terminals with serial number prefixes from 2420F onwards. If changes have been made to the terminal since this manual was printed, a yellow "manual updating supplement" defining them is supplied with this manual to explain how to adapt the manual to the modified equipment. The changes will be incorporated in the next revision of this manual.

### NOTICE

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

This document contains proprietary information which is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced or translated to another language without the prior written consent of Hewlett-Packard Company.

Hewlett-Packard  
Roseville Terminal Division  
8020 Foothills Blvd.  
Roseville, California U.S.A 95678

Manual Part Number: 02394-90001

Printed in Singapore

04/1985

(

(

(



# Table of Contents

Section I — General Description	Page		Page
INTRODUCTION .....	1-1	USER-DEFINABLE KEYS .....	3-11
THE HP 2392A FEATURE SET .....	1-1	Defining Keys Locally .....	3-12
THE HP 2394A FEATURE SET .....	1-2	Defining Keys Programmatically .....	3-13
KEYBOARD .....	1-8	Controlling The User Keys Menu	
FUNCTION KEYS AND SCREEN LABELS .....	1-9	Programmatically .....	3-15
STATUS LINE .....	1-14	Controlling the Function Key Labels	
CONFIGURING THE TERMINAL .....	1-14	Programmatically .....	3-15
TERMINAL PORTS .....	1-14	ENABLE/DISABLE KEYBOARD .....	3-16
COMPUTER PORT DATA		[Print Enter] KEY .....	3-16
COMMUNICATIONS .....	1-16	Enter .....	3-16
EXTERNAL PRINTER PORT .....	1-16	Print .....	3-16
SELF-TEST .....	1-16	SEND DISPLAY .....	3-17
 Section II — Configuring The Terminal		[Reset Break] KEY .....	3-23
INTRODUCTION .....	2-1	Soft Reset .....	3-23
CONFIGURATION FUNCTION KEYS .....	2-1	Hard Reset .....	3-23
TERMINAL CONFIGURATION MENU .....	2-2	Break .....	3-24
Lock/Unlock Configuration Menus .....	2-13	[Extend char] KEY .....	3-24
Setting Terminal Configuration Parameters		[Select] KEY .....	3-24
With Escape Codes .....	2-14	[Return] KEY .....	3-24
Terminal Configuration Parameters Unique		[Stop] KEY .....	3-25
to the 2394A .....	2-16	BELL .....	3-25
Datacomm Menu (HP 2394A) .....	2-19	WAIT .....	3-25
 Section III — Keyboard Control		MODEM DISCONNECT .....	3-25
INTRODUCTION .....	3-1	 Section IV — Alphanumeric Display Control	
SELECTING THE MODE .....	3-1	INTRODUCTION .....	4-1
Line Modify Mode .....	3-1	CURSOR CONTROL .....	4-2
Modify All Mode .....	3-2	Home Up .....	4-2
Character/Block Mode .....	3-3	Home Down .....	4-3
Remote/Local Mode .....	3-5	Move Cursor Up .....	4-3
Smooth Scroll Mode .....	3-5	Move Cursor Down .....	4-3
Memory Lock Mode .....	3-6	Move Cursor Right .....	4-4
Display Functions Mode .....	3-7	Move Cursor Left .....	4-4
Auto Line Feed Mode .....	3-8	Roll Text Up .....	4-4
Caps Mode .....	3-9	Roll Text Down .....	4-5
Caps Lock Mode .....	3-9	Next Page/Previous Page .....	4-5
Send Cursor Position Mode .....	3-10	MEMORY ADDRESSING SCHEME .....	4-6
Format Mode .....	3-10	Cursor Sensing .....	4-9
Auto-keyboard Lock Mode .....	3-11	Cursor Positioning .....	4-9

## Table of Contents (continued)

Page	Page
Screen Relative Addressing .....	4-10
Memory Absolute Addressing .....	4-10
Cursor Relative Addressing .....	4-11
Combining Absolute and Relative Addressing .....	4-12
EDIT OPERATIONS .....	4-13
Insert Line .....	4-13
Delete Line .....	4-14
Insert Character .....	4-14
Delete Character .....	4-16
Clear Display .....	4-17
Clear Line .....	4-18
SETTING AND CLEARING MARGINS .....	4-18
SETTING AND CLEARING TABS .....	4-19
FORWARD TAB .....	4-20
BACK TAB .....	4-20
FORMAT MODE TAB SEQUENCE	
CONTROL (2394A) .....	4-21
SCREEN BLANKING .....	4-26
DISPLAY ENHANCEMENTS .....	4-26
DISPLAY ENHANCEMENTS USING THE SYSTEM FUNCTION KEYS (HP 2394A) .....	4-29
DESIGNING AND USING FORMS .....	4-30
DRAWING FORMS USING THE LINE DRAWING CHARACTER SET .....	4-31
FORMS MODE (FORMAT MODE) .....	4-33
Protected Fields .....	4-33
Unprotected Fields .....	4-33
READING AND STORING FORMS .....	4-34
DESIGNING AND USING FORMS ON THE HP 2394A .....	4-34
Selecting Line Types .....	4-34
Setting Margins .....	4-35
Drawing Lines .....	4-35
Protected and Unprotected Fields (2394A) ...	4-40
Transmit-Only Fields (2394A) .....	4-40
FIELD ATTRIBUTES (2394A) .....	4-40
DEFINING FIELDS FROM THE KEYBOARD (2394A) .....	4-43
TRANSMIT MODIFIED FIELDS (2394A) .....	4-48
Setting Modified Data Tags .....	4-48
Configuring the Terminal .....	4-48
FORMS CACHE (STORAGE) (2394A) .....	4-49
Selecting Forms Cache Memory .....	4-49
Defining a Form .....	4-49
Purging a Form .....	4-51
Transferring a Form to the Screen .....	4-51
Forms Cache Status .....	4-51
Reading the Forms Cache Directory .....	4-52
Forms Cache Internals .....	4-53
SUMMARY (2394A) .....	4-55
<b>Section V — External Printer Control</b>	
INTRODUCTION .....	5-1
Record Mode .....	5-1
DATA LOGGING .....	5-3
Top Logging .....	5-3
Bottom Logging .....	5-3
DISPLAY TO DEVICE ALPHANUMERIC DATA TRANSFERS .....	5-4
Advance Page .....	5-4
Advance Line .....	5-4
Copy All .....	5-5
Format Mode (2394A) .....	5-5
Copy Page .....	5-5
Copy Line .....	5-5
Copy Display Memory .....	5-6
Copy Menu .....	5-6
Device Control Completion Codes .....	5-6
COMPUTER TO DEVICE DATA TRANSFERS .....	5-7

# Table of Contents (continued)

	Page		Page
CONFIGURING THE EXTERNAL		TERMINAL SELF-TESTS .....	8-3
PRINTER .....	5-9	Power-on Test .....	8-3
Printer Cabling .....	5-9	Terminal Test .....	8-4
Filling in the Configuration Menu .....	5-11	Identify ROMS .....	8-5
		Port 1 Test .....	8-6
		Port 2 Test .....	8-7
<b>Section VI — Data Communications</b>		<b>Section IX — Terminal Maintenance</b>	
INTRODUCTION .....	6-1	<b>Procedures</b>	
HARDWIRE CONNECTIONS .....	6-2	INTRODUCTION .....	9-1
MODEM CONNECTIONS .....	6-5	CLEANING THE SCREEN AND	
Modem Considerations .....	6-5	KEYBOARD .....	9-1
INSTALLING A POINT-TO-POINT		BATTERY MAINTENANCE .....	9-1
CONFIGURATION .....	6-7		
CONFIGURING THE TERMINAL .....	6-7		
POINT-TO-POINT PROGRAMMING			
INFORMATION .....	6-11	<b>Section X — ANSI Operation</b>	
Start and Stop Bits .....	6-11	INTRODUCTION .....	10-1
Parity Checking .....	6-11	ANSI/EM52/HP MODE	
Receive Buffer .....	6-12	CONFIGURATION .....	10-3
Receive Errors .....	6-12	MODES OF OPERATION .....	10-3
Local/Remote Mode .....	6-13	HP Mode .....	10-3
Full-duplex Operation .....	6-13	EM52 Mode .....	10-3
Pacing Mechanism .....	6-14	ANSI Mode .....	10-4
Transfers and Handshaking .....	6-15	ESCAPE SEQUENCES .....	10-5
Data Transfer Priority .....	6-16	HP Mode .....	10-5
		EM52 Mode .....	10-5
<b>Section VII — Status</b>		ANSI Mode .....	10-5
INTRODUCTION .....	7-1	Escape Sequences Generated	
INTERPRETING STATUS .....	7-2	By The Keyboard .....	10-6
Terminal ID Status .....	7-2	Selecting Terminal Mode	
Terminal Status .....	7-3	Programmatically .....	10-6
TERMINAL CAPABILITIES (2394A) .....	7-12	CURSOR CONTROL KEYS .....	10-7
DEVICE STATUS .....	7-17	NUMERIC KEYPAD .....	10-8
		ALTERNATE CHARACTER SETS .....	10-9
<b>Section VIII — Error Messages</b>		HP Mode .....	10-9
<b>And Self-Test</b>		EM52 Mode .....	10-10
INTRODUCTION .....	8-1	ANSI Mode .....	10-10
ERROR MESSAGES .....	8-1	ANSI CONFIGURATION MENU .....	10-11

# Table of Contents (continued)

	Page		Page
Multipage Field .....	10-13	PP - Previous Page .....	10-24
Backspace Definition Field .....	10-13	SD - Scroll Down .....	10-25
Setting And Saving Tabs .....	10-14	SU - Scroll Up .....	10-25
Answerback Message Field .....	10-14	HPSTBM - Set Top And	
ANSI CONTROL SEQUENCE		Bottom Margins .....	10-26
SUMMARY .....	10-15	ANSI Editing Control Sequences .....	10-26
Control Characters .....	10-16	DCH - Delete Character .....	10-26
ANSI Cursor Control Sequences .....	10-18	DL - Delete Line .....	10-26
CBT - Cursor Back Tabulation .....	10-18	ED - Erase in Display .....	10-27
CHA - Cursor Horizontal Absolute .....	10-19	EL - Erase in Line .....	10-27
CNL - Cursor Next Line .....	10-19	IL - Insert Line .....	10-27
CPL - Cursor Preceding Line .....	10-19	ANSI Character Set Selection Sequences ..	10-27
CPR - Cursor Position Report .....	10-19	SCS - Select Character Set .....	10-28
CUB - Cursor Backward .....	10-19	SGR - Select Graphic Rendition .....	10-28
CUD - Cursor Down .....	10-19	HPDHL - Double Size Line .....	10-28
CUF - Cursor Forward .....	10-20	HPDWL - Double Width Line .....	10-29
CUP - Cursor Position .....	10-20	HPSSL - Single Size Line .....	10-29
CUU - Cursor Up .....	10-20	ANSI Terminal Status Sequences .....	10-29
HTS - Horizontal Tab Set .....	10-20	DA - Device Attributes .....	10-29
HVP - Horizontal and		DSR - Device Status Report .....	10-30
Vertical Position .....	10-20	HPID - Identify .....	10-30
IND - Index .....	10-21	HPREQTPARM - Request Terminal	
NEL - Next Line .....	10-21	Parameters .....	10-30
RI - Reverse Index .....	10-21	HPRPTARM - Report Terminal	
TBC - Tab Clear .....	10-22	Parameters .....	10-31
VPA - Vertical Position Absolute .....	10-22	ANSI Terminal Control Sequences .....	10-31
VPR - Vertical Position Relative .....	10-22	MC - Media Copy .....	10-31
HPA - Horizontal Position Absolute .....	10-22	RIS - Reset To Initial State .....	10-31
HPR - Horizontal Position Relative .....	10-23	HPKPAM - Keypad	
HPHD - Home Down .....	10-23	Application Mode .....	10-32
HPHU - Home Up .....	10-23	HPKPNM - Keypad Numeric Mode .....	10-32
HPSC - Save Cursor and attributes .....	10-23	HPLF - Load Flags .....	10-32
HPRC - Restore Cursor and		ANSI Terminal Mode	
attributes .....	10-23	Selection Sequences .....	10-33
ANSI Display Control Sequences .....	10-24	RM - Reset Mode .....	10-34
NP - Next Page .....	10-24	SM - Set Mode .....	10-34

# Table of Contents (continued)

	Page		Page
KAM - Keyboard Action Mode .....	10-34	ESCAPE SEQUENCE SUMMARY .....	10-41
CRM - Control Representation Mode ...	10-34	ANSI Escape Sequences .....	10-41
IRM - Insert/Replace Mode .....	10-35	EM52 Escape Sequences .....	10-43
LNM - New Line Mode .....	10-35	<b>Appendix A — Escape Codes</b>	
HPCKM - HP cursor Control		TERMINAL CONTROL FUNCTIONS .....	A-1
Keys Mode .....	10-35	CURSOR CONTROL OPERATIONS .....	A-4
HPANM - HP/ANSI/EM52 Mode .....	10-36	CONFIGURATION OPERATIONS .....	A-5
HPSCLM - HP Scrolling Mode .....	10-36	DATA TRANSFER OPERATIONS .....	A-6
HPOM - HP Origin Mode .....	10-36	FORMS CACHE (STORAGE) (2394A) .....	A-8
HPAWM - HP Autowrap Mode .....	10-37	FORMAT MODE .....	A-9
HPMPM - HP Multipage Mode .....	10-37	FUNCTION KEY AND ERROR MESSAGE	
HPMLM - HP Memory Lock Mode ....	10-37	OPERATIONS .....	A-10
EM52 CONTROL SEQUENCE		CONTROL CHARACTERS .....	A-13
SUMMARY .....	10-38	TERMINAL CONFIGURATION MENU	
Cursor Up .....	10-38	OPERATIONS .....	A-14
Cursor Down .....	10-38	DATACOMM MENU .....	A-16
Cursor Right .....	10-38	DISPLAY ENHANCEMENTS	
Cursor Left .....	10-38	OPERATIONS .....	A-12
Cursor to Home .....	10-38	<b>Appendix B — Character Set and Keyboards</b>	
Reverse Line Feed .....	10-39	CHARACTER SETS .....	B-1
Erase To End Of Screen .....	10-39	7-Versus 8 Bit Transmission to the	
Erase To End Of Line .....	10-39	Computer .....	B-6
Direct Cursor Address .....	10-39	Data Transfers to the External Printer .....	B-7
Identify .....	10-39	KEYBOARDS .....	B-7
Enter Alternate Keypad Mode .....	10-40	Diacriticals Characters .....	B-8
Exit Alternate Keypad Mode .....	10-40	LANGUAGES .....	B-9
Enter ANSI Mode .....	10-40		

# Illustrations

Figure	Page	Figure	Page
1-1	2392A/2394A Terminals . . . . .	1-4	
1-2	Standard USASCII Keyboard . . . . .	1-8	
1-3	Function Keys and Screen Labels (2392A) . . . . .	1-9	
1-4	Function Keys and Screen Labels (2394A) . . . . .	1-10	
1-5	Mode Selection Labels . . . . .	1-11	
1-6	Function Key Hierarchy of the 2392A . .	1-12	
1-7	Function Key Hierarchy of the 2394A . .	1-13	
1-8	User Keys Definition Menu . . . . .	1-15	
1-9	Default User Key Labels . . . . .	1-15	
1-10	Sample User-defined User Key Labels . .	1-15	
1-11	Screen Status Line . . . . .	1-15	
2-1	Terminal Configuration Menu of the 2392A . . . . .	2-2	
2-2	Terminal Configuration Menu of the 2394A . . . . .	2-3	
3-1	Screen-labeled Function Keys (2392A) . .	3-3	
3-2	Screen-labeled Function Keys (2394A) . .	3-4	
3-3	User Keys Definition Menu and Default User Key Labels . . . . .	3-13	
4-1	The "Roll" Data Functions . . . . .	4-5	
4-2	Previous Page and Next Page Concepts .	4-7	
4-3	Row Addressing . . . . .	4-8	
4-4	Column Addressing . . . . .	4-8	
4-5	Character Insert With Margins . . . . .	4-15	
4-6	Character Delete With Margins . . . . .	4-16	
4-7	Vertical Range Tab Sequence Ex (2394A) . . . . .	4-22	
4-8	Top Horizontal Tab Sequence Ex (2394A) . . . . .	4-23	
4-9	Bottom Horizontal Tab Sequence Ex (2394A) . . . . .	4-24	
4-10	Mid-Range Tab Sequence Ex (2394A) . .	4-25	
4-11	Menu After Entering Four Enhancements Escape Sequences . . . . .	4-28	
4-12	Line Drawing Set Elements . . . . .	4-31	
4-13	Sample Data Entry Form . . . . .	4-32	
4-14	Typical Form . . . . .	4-33	
4-15	Sample Hardcopy Form . . . . .	4-36	
4-16	Sample Hardcopy Form . . . . .	4-37	
4-17	Field Definition Menu (2394A) . . . . .	4-44	
4-18	Typical Data Entry Form . . . . .	4-55	
5-1	External Device Port Cabling . . . . .	5-10	
5-2	External Device Configuration Menu . . .	5-12	
6-1	Terminal Rear View . . . . .	6-6	
6-2	Terminal Cabling . . . . .	6-6	
6-3	Datacomm Configuration Menu . . . . .	6-8	
7-1	Primary Terminal Status Example . . . . .	7-4	
7-2	Secondary Terminal Status Example . . . .	7-8	
7-3	Terminal Capabilities Status Example (2394A) . . . . .	7-12	
7-4	Terminal Alphanumeric Status Bytes (2394A) . . . . .	7-13	
7-5	Terminal Graphics Status Bytes (2394A) . . . . .	7-14	
7-6	Installed Memory Status Bytes (2394A) . . . . .	7-15	
7-7	Terminal Interface Capabilities Status Bytes (2394A) . . . . .	7-16	
7-8	External Device Status Byte . . . . .	7-17	
8-1	Screen Test Pattern . . . . .	8-5	
8-2	ROM Identification Listing . . . . .	8-5	
9-1	Removing The Batteries . . . . .	9-2	

## Illustrations (continued)

Figure	Page	Figure	Page
10-1	Terminal Configuration Menu . . . . .	10-2	
10-2	HP Mode Screen Status Line . . . . .	10-3	
10-3	EM52 Mode Screen Status Line . . . . .	10-3	
10-4	ANSI Mode Screen Status Line . . . . .	10-4	
10-5	Numeric keypad Overlay . . . . .	10-10	
10-6	HP Line Drawing Set Elements . . . . .	10-11	
10-7	ANSI Line Drawing Set Elements . . . . .	10-11	
10-8	ANSI Configuration Menu . . . . .	10-12	
B-1	Accessing ROMAN 8 and Line Drawing Set . . . . .	B-1	
B-2	Line Drawing Set Elements . . . . .	B-6	
B-3	Mapping Of ROMAN 8 Characters (accessed by the [Extend char] key) . . .	B-9	
B-4	American USASCII Keyboard . . . . .	B-9	
B-5	Swedish Keyboard . . . . .	B-10	
B-6	Norwegian Keyboard . . . . .	B-10	
B-7	French Keyboard . . . . .	B-10	
B-8	German Keyboard . . . . .	B-11	
B-9	English (UK) Keyboard . . . . .	B-11	
B-10	Spanish (European) Keyboard . . . . .	B-11	
B-11	Canadian (French) Keyboard . . . . .	B-12	
B-12	Canadian (English) Keyboard . . . . .	B-12	
B-13	Italian Keyboard . . . . .	B-12	
B-14	Dutch Keyboard . . . . .	B-13	
B-15	Finnish Keyboard . . . . .	B-13	
B-16	Danish Keyboard . . . . .	B-13	
B-17	Swiss (German) Keyboard . . . . .	B-14	
B-18	Swiss (French) Keyboard . . . . .	B-14	
B-19	Spanish (Latin American) Keyboard . . .	B-14	
B-20	Flemish Keyboard . . . . .	B-15	

# Tables

Table	Page	Table	Page
1-1 Specifications .....	1-5	7-1 Status Characters .....	7-3
2-1 Terminal Configuration Menu Fields ...	2-4	10-1 HP And EM52 Control Sequences Generated By Cursor Keys .....	10-7
2-2 Configuration Function Keys [f4], [f7], [f8] .....	2-13	10-2 ANSI Control Sequences Generated By Cursor Keys .....	10-8
3-1 [Enter] Key Operation .....	3-18	10-3 ANSI And EM52 Control Sequences Generated By Numeric Keypad .....	10-9
4-1 Display Enhancement Character .....	4-27	10-4 ANSI/EM52 Mode Control Character Recognition .....	10-17
4-2 Line Drawing Function Keys (2394A) ..	4-39	10-5 HPRTPAR Report Parameters Mnemonics .....	10-30
4-3 Attributes by Field Type (2394A) .....	4-41	10-6 ANSI And HP Private Modes .....	10-33
4-4 <define fields> Function Keys (2394A) .....	4-45		
5-1 External Device Port Data Communications Cables .....	5-10	B-1 ROMAN 8 Characters Supported By 2392A/2394A .....	B-2
5-2 External Device Configuration Menu Fields .....	5-13	B-2 Non-displayable Control Character Set ..	B-3
		B-3 National Characters Replaced By USASCII Characters in 7-bit Mode .....	B-4
6-1 Data Communications Cables And Test Connectors .....	6-3	B-4 Diacritical Marks in 8-bit Operation ....	B-5
6-2 Datacomm Configuration Menu Fields ..	6-8	B-5 Extended Roman Data Transfers To Printer .....	B-7
6-3 DC1/DC2 Handshake Used In Data Transfers .....	6-16		



# Preface

The HP 2392A Display Terminal is a versatile, ergonomic, character, line, or block mode CRT terminal. The HP 2394A Data Entry Terminal has all the features of the HP 2392A Display Terminal PLUS increased capability with respect to forms design and forms handling.

This manual contains information on both the HP 2392A Display Terminal and the HP 2394A Data Entry Terminal. The 2394A Data Entry Terminal has all the features of the HP 2392A Display Terminal plus enhanced data entry features. The enhanced features of the HP 2394A terminal are pointed out throughout the manual by listing the product number (HP 2394A) next to the feature or by footnotes. A feature comparison between the two products appears on page XI. The additional features are also noted in the index in the same manner.

This reference manual contains detailed information for configuring, testing, and using these terminals.

For ease of use, the manual is divided into nine sections and two appendices, as follows:

Section I	General Description
Section II	Configuring the Terminal
Section III	Keyboard Control
Section IV	Alphanumeric Display Control
Section V	External Printer Control
Section VI	Data Communications
Section VII	Status
Section VIII	Error Messages and Self-Test
Section IX	Terminal Maintenance Procedures
Section X	ANSI Operation
Appendix A	Escape Codes
Appendix B	Character Sets and Keyboards

## Comparison of the HP 2392A and the HP 2394A

FEATURES	HP 2392A	HP 2394A
Display		
Size	12 "	12 "
Format:		
Columns	80	80
Lines	24	24
Softkeys/Labelling Status	25th, 26th 27th	25th, 26th 27th
Character Cell	9 × 14	9 × 14
Video Enhancements <sup>1</sup>	UL, IV, B, HB, SV	UL, IV, B, HB, SV
Phosphor Memory	Green Up to 4 pages std. Up to 8 pages opt.	Green Up to 8 pages std.
Total Memory (Std.)	16K RAM	48K RAM
Forms Cache	No	Yes
Forms Cache Directory	No	Yes
Field Attributes <sup>2</sup>	P, UNP	P, UNP, T-0
Modified Data Tags	No	Yes
Forms Tab Control	No	Yes
Data Edit Checks	No	11 Checks
Line Drawing Set	Yes	Yes
Keyboard		
Typewriter	Yes	Yes
Detached	Yes	Yes
Numeric Pad	Yes	Yes
Cursor Keys	Yes	Yes
Editing Keys <sup>3</sup>	IC, DC, IL, DL	IC, DC, IL, DL
Function Keys		
Fixed Definition	None	8
User Definable	8	8
Menu Aids for Forms Design		
Video Enhancements Menu	No	Yes
Field Attributes	No	Yes
Local Edits Menu	No	Yes
Sketch Forms Menu	No	Yes
Select, Return, and Tab Keys	Fixed	Programmable

**HP 2392A/HP 2394A Comparison (continued)**

<b>FEATURES</b>	<b>HP 2392A</b>	<b>HP 2394A</b>
Data Communications		
Mode:		
Character/Line	Yes	Yes
Block	Yes	Yes
Remote Configuration via ESC Sequences	No	Yes
Speed	110-19200 baud	110-19200 baud
Interface:		
RS232	Std.	Std.
HP422	Std.	Std.
X.25 PAD	Std.	Std.
Protocol — ENQ/ACK, X ON/OFF	Yes	Yes
ANSI Capability	No cost option	No cost option
Hard Copy		
Printer Ports:		
RS232	Optional	Optional
Centronix-type parallel	Optional	Optional

**Keys:**

- <sup>1</sup> UL=Underline IV=Inverse Video B=Blinking HB=Halfbright SV=Security Video  
<sup>2</sup> P=Protected UNP=Unprotected  
<sup>3</sup> IC=Insert Character DC=Delete Character IL=Insert Line DL=Delete Line

C

C

C

# General Description

SECTION

I

## INTRODUCTION

### The HP 2392A Feature Set

The HP 2392A Feature Set\* (figure 1-1) is a versatile general purpose alphanumeric CRT terminal that offers the following features:

- CRT screen size of 170 mm (6.7 inches) by 250 mm (9.8 inches).
- Screen capacity of 24 lines of up to 80 characters each (1920 characters total). Plus three additional lines; two for function key labels, and one for terminal status.
- Display enhancements (inverse video, blinking, underlining, half-bright and security).
- Character, line, or block mode operation.
- Eight user-definable function keys ([f1] through [f8]) are provided. The function of these keys is displayed in two rows of labels across the bottom of the display screen. The labels are definable (up to 16 characters each). The character strings returned when the keys are pressed can be defined with up to 80 characters.
- All terminal configuration operations are performed through keyboard entries into formatted menus displayed on the screen, there are no physical straps. (Configuration data is maintained in non-volatile memory).
- Two-character user-definable [Return] key.
- Line drawing character set.
- Screen labeled system function keys (for selecting operating modes and performing other terminal control functions).
- Extended Character Set (ROMAN 8, supports national keyboard layouts).
- Full editing capabilities (insert/delete/clear line and insert/delete character).
- Adjustable margins and tab stops.
- Programmatic cursor sensing and addressing.
- Easy-to-use keyboard with numeric key pad.
- Extensive self-test capability.

Table 1-1 lists the terminal specifications.

\*see page 1-2 for the extended capability of the HP 2394A Data Entry Terminal.

## The HP 2394A Feature Set

The HP 2394A Data Entry Terminal has all the features of the HP 2392A Display Terminal plus:

- 48K RAM (16K more than max. in the HP 2392A)
- 96K ROM (16K more than the HP 2392A)
- Local Forms Cache capability enables the system to store forms locally in the terminal. Rather than transmitting a form to the terminal each time it is needed, forms to be used in a work session can be sent to the terminal at the beginning of the session. Forms can then be transferred locally from forms cache to the screen. A Forms Cache Directory allows an application program to determine exactly which forms are currently stored in forms cache.
- Local Edit Checking is performed by the terminal (format mode) to insure that data is in the specified format before it is returned to the host computer. The following is a list of these edit checks.
  - All characters (default)
  - Alphabetic-only
  - Auto Upshift
  - Alphanumeric
  - Integer-only
  - Signed decimal
  - Implied decimal
  - Constant
  - Integer with zero fill
  - Signed decimal with fill
  - Implied decimal with fill
  - Numeric only
  - Justify
  - Required field
  - Total fill
  - Permanent modified data tag
- Select number of implied decimals (for edits)
- Select decimal type (U.S. or European; for edits)
- Modified Data Tag allows the user to send only those fields in a form which have been modified. This can greatly reduce data communications traffic as well as reducing system overhead, particularly, when used in conjunction with local edits.
- Transmit Only Fields
- Menu Aids for Form Design
  - Defining Unprotected/Transmit Only fields
  - Defining video enhancements
  - Defining edits
  - Defining lines (with line drawing set)

- Format Mode Tab (sequence) Control allows an application to define the tab sequence with respect to unprotected fields. This capability allows the tab key to access unprotected fields in a logical sequence even though they may be grouped vertically.
- 8 user definable function keys plus
- 8 function keys with set definitions.
- Additional configuration options
  - return = enter key
  - tab = spaces
- Modify Permanent Configuration Menus via escape sequences. This feature could be used to reconfigure the terminal remotely to talk to different systems.
- Optional video enhancements of user softkey labels
- Programmatic triggering of Return key
- Select key disable.

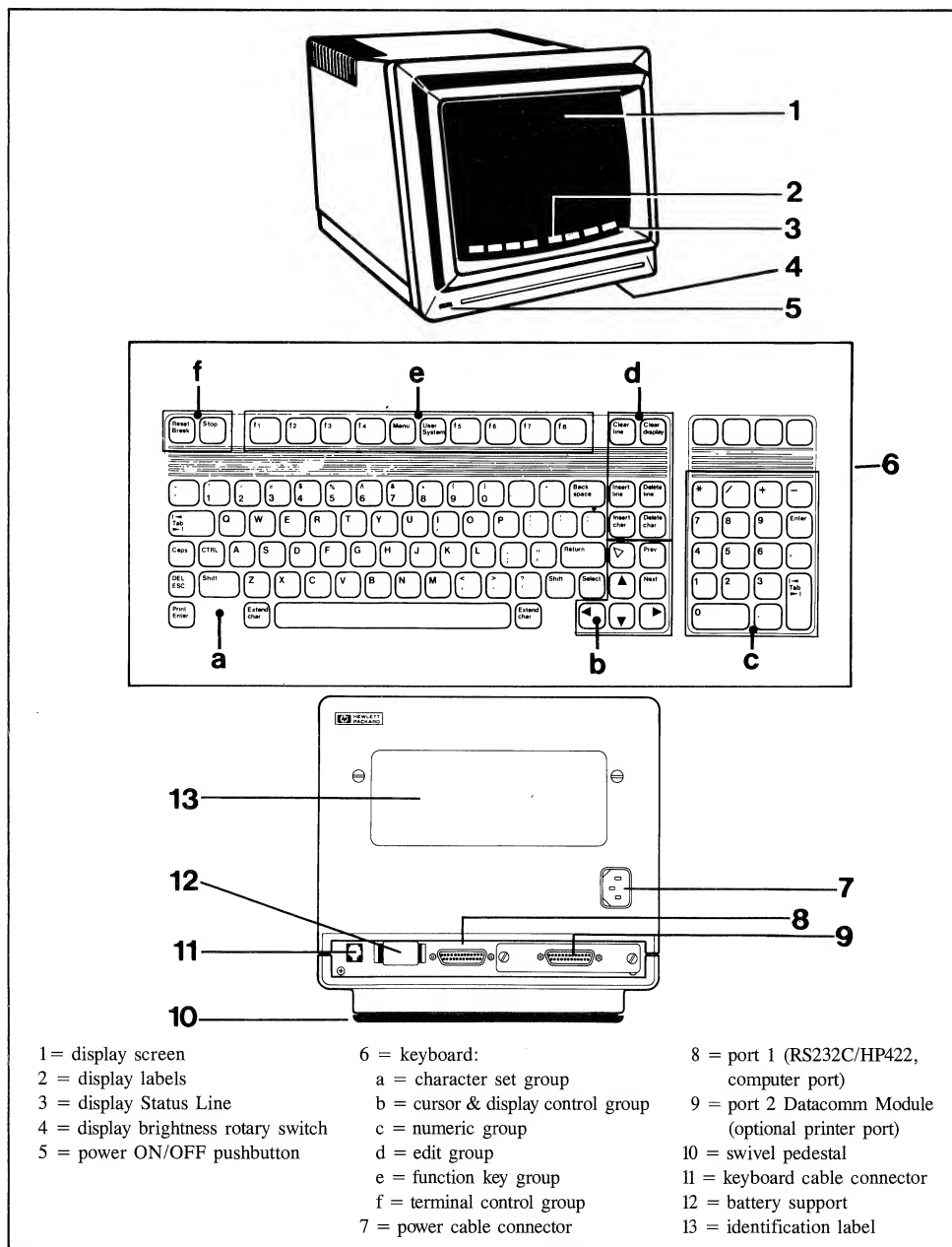


Figure 1-1. 2392A Display Terminal/2394A Data Entry Terminal



Table 1-1. Specifications

<b>GENERAL</b> (Standard terminal)	
Screen Size:	170 mm (6.7 inches) by 250 mm (9.8 inches).
Screen Capacity:	24 lines of 80 columns (1920 characters). Two additional lines for function key labels, plus a status line.
Character Generation:	7 by 11 character in 9 by 14 dot cell.
Character Size:	2.4 mm (0.094 inch) by 3.5 mm (0.138 inch).
Character Set:	ROMAN 8 (comprising 128 ASCII character set and extended Roman set) and Line Drawing.
Cursor:	Blinking-underline.
Display Enhancements:	Inverse, underline, blinking, half-bright, security.
Display Memory:	
HP 2392A:	Up to four pages of memory (16K RAM)
HP 2394A:	Up to eight pages of memory (32K RAM) A minimum of 16K is allocated to forms cache Total memory = 48K RAM
Display Modes:	Light characters on dark background.
Refresh Rate:	60 Hz.
Keyboard:	Detached, with 1.2 m (4 feet) coiled cable. 8 screen labeled keys; auto-repeat; N-key rollover. Numeric keyboard and independent user and cursor keys.
Operating Modes:	Local/remote Character/block (line/page) Forms/non-forms. Forms cache (HP 2394A only)
Transmission Modes:	Full duplex, asynchronous point-to-point. Optional via full duplex modems.
Electrical Interface:	RS232C/HP 422.
Data Rates:	110, 134.5, 150, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200 baud.
Parity:	Selectable: Even, odd, zero, one, none.

Continued . . .

Table 1-1. Specifications (Continued)

**PHYSICAL CHARACTERISTICS**

Weight:	Display Monitor:
	10.8 kg (23.8 lbs)
	Keyboard:
	2.2 kg (4.9 lbs)
Dimensions:	Display Monitor:
	325 mm wide by 362 mm deep by 317 mm high
	(12.8 inches by 14.2 inches by 12.5 inches)
	Keyboard:
	455 mm wide by 220 mm deep by 40 mm high
	(17.9 inches by 8.7 inches by 1.6 inches)

**POWER REQUIREMENTS**

Input Voltage:	115 V (+ 10%, - 25%) at 47-66 Hz
	230 V (+ 10%, - 15%) at 47-66 Hz
Power Consumption:	50 watts (standard)

**HP 2392A/HP 2394A OPTIONS**

015	230 V 50 Hz operation	
049	ANSI X 3.64 operation	
092	25-pin RS232C datacomm port	} User installable port 2 connector
093	8-bit parallel datacomm port	
101	Swedish	} National language keyboards
102	Norwegian	
103	French	
104	German	
105	English, UK	
106	Spanish, European	
107	Canadian, French	
108	Canadian, English	
109	Italian	
110	Dutch	
111	Finnish	
112	Danish	
113	Swiss, German	
114	Swiss, French	
115	Spanish, Latin American	
116	Flemish	
149	Delete keyboard	
160	Memory extension (adds an additional four pages of display memory) *	
301	US modem cable (same as accessory 40242M)	} Computer cables
302	European modem cable (same as 40242M)	
303	RS232C cable (same as 40242C)	

\*This additional 16K of RAM is only available on the HP 2392A.

Continued . . .

Table 1-1. Specifications (Continued)

304	HP direct connect type 232 cable (same as 40242X)	} Computer cables
305	EMP protect cable (same as 40242Y)	
306	HP direct connect type 422 cable (same as 40242P)	
<b>HP 2392A/HP 2394A Accessories</b>		
40210M	Current loop datacomm module	
40210P	8-bit parallel datacomm module	
40210R	25-pin RS232C datacomm module	
40242A	Combined RS232C and HP422 adapter. Male (25-pin)/female (50-pin), 1 m (3 ft)	
40242C	RS232C cable. Male (25-pin)/female (25-pin), 5 m (16 ft)	
40242M	US/European modem cable. Male (25-pin)/male (25-pin), 5 m (16 ft)	
40242P	HP direct connect type 422 cable. Male (25-pin)/male (5-pin), 5 m (16 ft)	
40242X	HP direct connect type 232 cable. Male (25-pin)/male (3-pin), 5 m (16 ft)	
40242Y	EMP protect cable. Male (25-pin)/male (25-pin), 5 m (16 ft)	
40242Z	RS232C modem bypass cable. Male (25-pin)/female (25-pin), 5 m (16 ft)	
13232U	Modem bypass cable. Female (25-pin)/female (25-pin), 1.5 m (5 ft)	
Note: The 40242 type cables provided RFI filtering, and must be used to comply with the FCC and VDE notices at the front of this manual.		
Cables used with 40242A		
13222C	RS232C cable. Male (50-pin)/female (25-pin), 2 m (6.6 ft)	
13222M	European modem cable. Male (50-pin)/male (25-pin), 5 m (16 ft)	
13222N	US modem cable. Male (50-pin)/male (25-pin), 5 m (16 ft)	
13222P	HP direct connect type 422 cable. Male (50-pin)/male (5-pin), 5 m (16 ft)	
13222W	RS232C cable. Male (25-pin)/female (50-pin), 5 m (16 ft)	
13222X	HP direct connect type 232 cable. Male (50-pin)/male (3-pin), 5 m (16 ft)	
13222Y	EMP protect cable. Male (25-pin)/male (50-pin), 5 m (16 ft)	
See Section V for printer cables		
46010A	American USASCII	} Keyboards
46010AC	Canadian, French	
46010AD	German	
46010AE	Spanish, European	
46010AF	French	
46010AH	Dutch	
46010AL	Canadian, English	
46010AM	Spanish, Latin American	
46010AN	Norwegian	
46010AQ	Swiss, French	
46010AR	Swiss, German	
46010AS	Swedish	
46010AU	English, UK	
46010AW	Flemish	
46010AX	Finnish	
46010AY	Danish	
46010AZ	Italian	

## KEYBOARD

The terminal keyboard is divided into five major groups of keys.

**Alphanumeric Group** – This group of keys is similar to a standard typewriter keyboard and consists of the alphabetic, numeric, and symbol keys. Included are lower and upper case alphabetic characters, control codes, punctuation characters, and some commercial symbols.

**Numeric Pad Group** – The numeric group of keys is located to the right of the alphanumeric keys. The layout of the numeric key pad is similar to that of a standard office calculator. These keys may be used for high-speed entry of large quantities of numeric data.

**Cursor Control Group** – This group of keys is used for moving the cursor around on the screen (up, down, left, or right) and for controlling what portion of the display appears on the screen (home up, home down, roll up, roll down, next page, and previous page).

**Edit Control Group** – These keys are used for inserting and deleting characters and lines in relation to the current cursor position.

**Function Group** – This group of keys ([F1] through [F8]) perform different functions depending upon which keystrokes have been performed. At any given time the applicable labels for these keys appear across the bottom of the display screen.

**Terminal Control Group** – This group of keys is used to interrupt terminal-to-computer communications, and initialize the terminal.

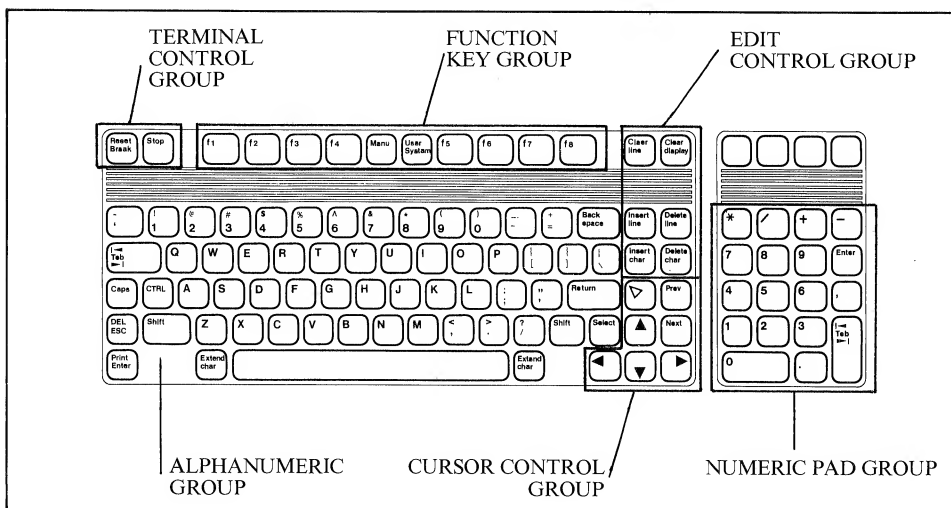


Figure 1-2. Standard USASCII Keyboard

The United States (USASCII) keyboard is the standard keyboard (see Figure 1-2). As an option you can order any of the national keyboards listed in Table 1-1. (The standard terminal is equipped with the ROMAN 8 character set that allows national characters to be accessed).

## FUNCTION KEYS AND SCREEN LABELS

Across the top of the keyboard are eight keys labeled [f1] through [f8] (Figure 1-3)\*. The functions performed by these keys change dynamically as you use the terminal. At any given time the applicable function labels for these keys appear across the bottom of the display screen.

When you press the [User System] key, the eight function keys become general control keys that you use for configuring the terminal, setting and clearing margins and tab stops, accessing the Service Keys, accessing the terminal modes keys, and accessing the Device Control group of keys. The entire set of system function key labels for each terminal is illustrated in Figures 1-6 and 1-7. Pressing [User System] always reinitializes [f1] through [f8] to the top row of functions (labels) shown in Figures 1-6 and 1-7.

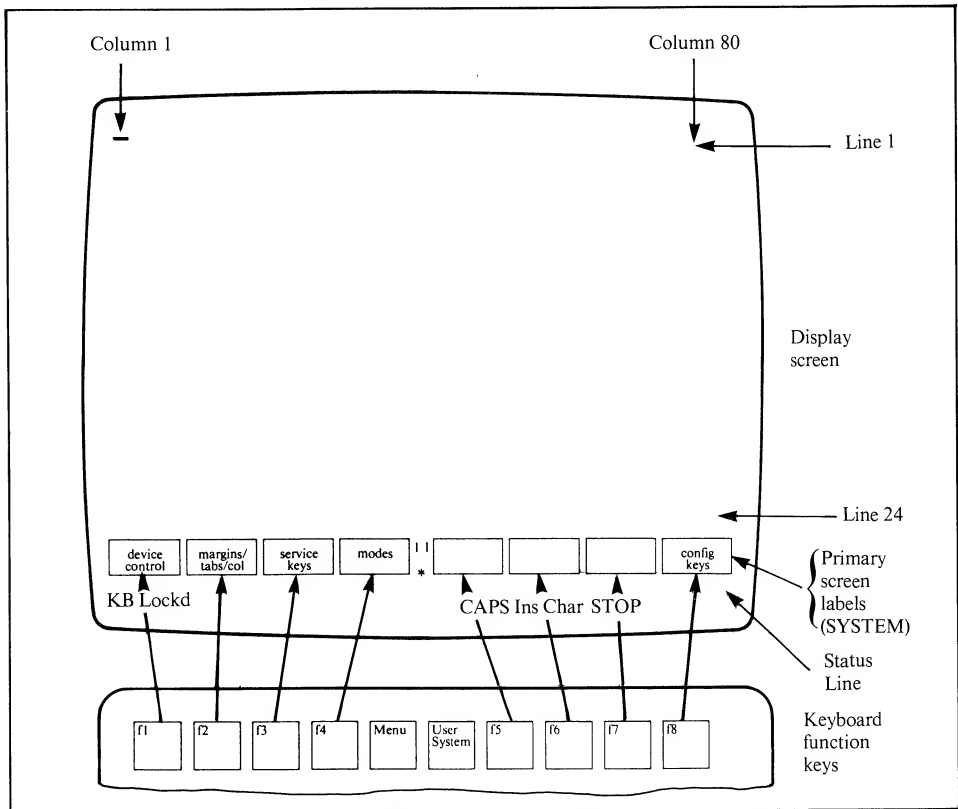


Figure 1-3. Function Keys and Screen Labels for the HP 2392A

\*See Figure 1-4 for the Function Keys and Screen Labels of the HP 2394A.

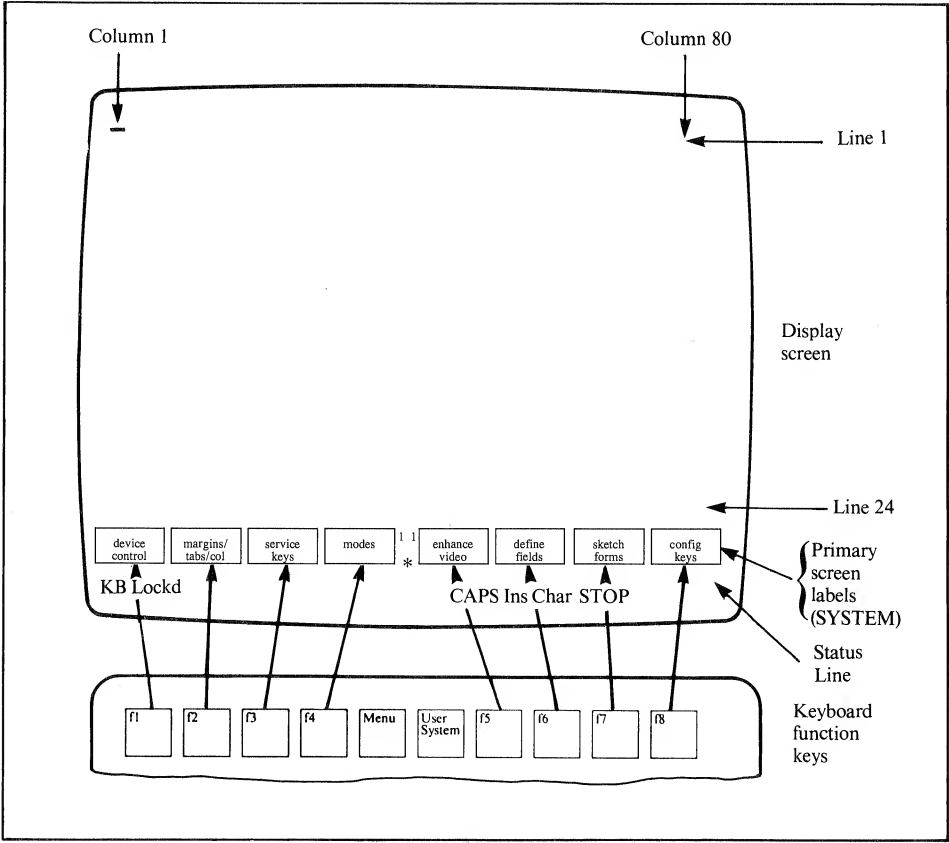


Figure 1-4. Function Keys and Screen Labels of the HP 2394A

In using the system function keys, keep in mind the following two conventions:

1. If a key label contains any lowercase letters, pressing the key will transfer you to another level of system function key labels.
2. If a key label contains only uppercase letters, pressing the key will perform the function defined in the key label.

For example, the key corresponding to label **AUTO LF** sets the automatic line feed function on; whereas the key corresponding to **config. keys** transfers you to the configuration function keys.

When you press the [User System], then [f4] [Modes] key, the eight function keys become mode selection keys (Figure 1-5). In this capacity you may use the keys to enable and disable various terminal operating modes (such as remote mode and display functions mode). Each mode selection key alternately enables and disables a particular mode. When the mode is enabled, an asterisk appears in the associated key label on the screen. At power-on, [f1] through [f8] are automatically initialized as mode selection keys.

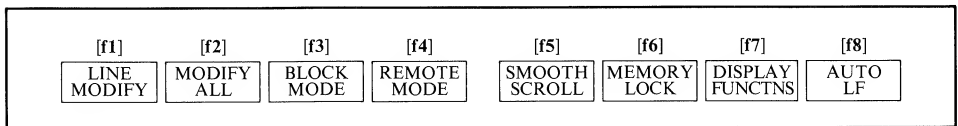


Figure 1-5. Mode Selection Labels

## HP 2392A Screen Label Tree

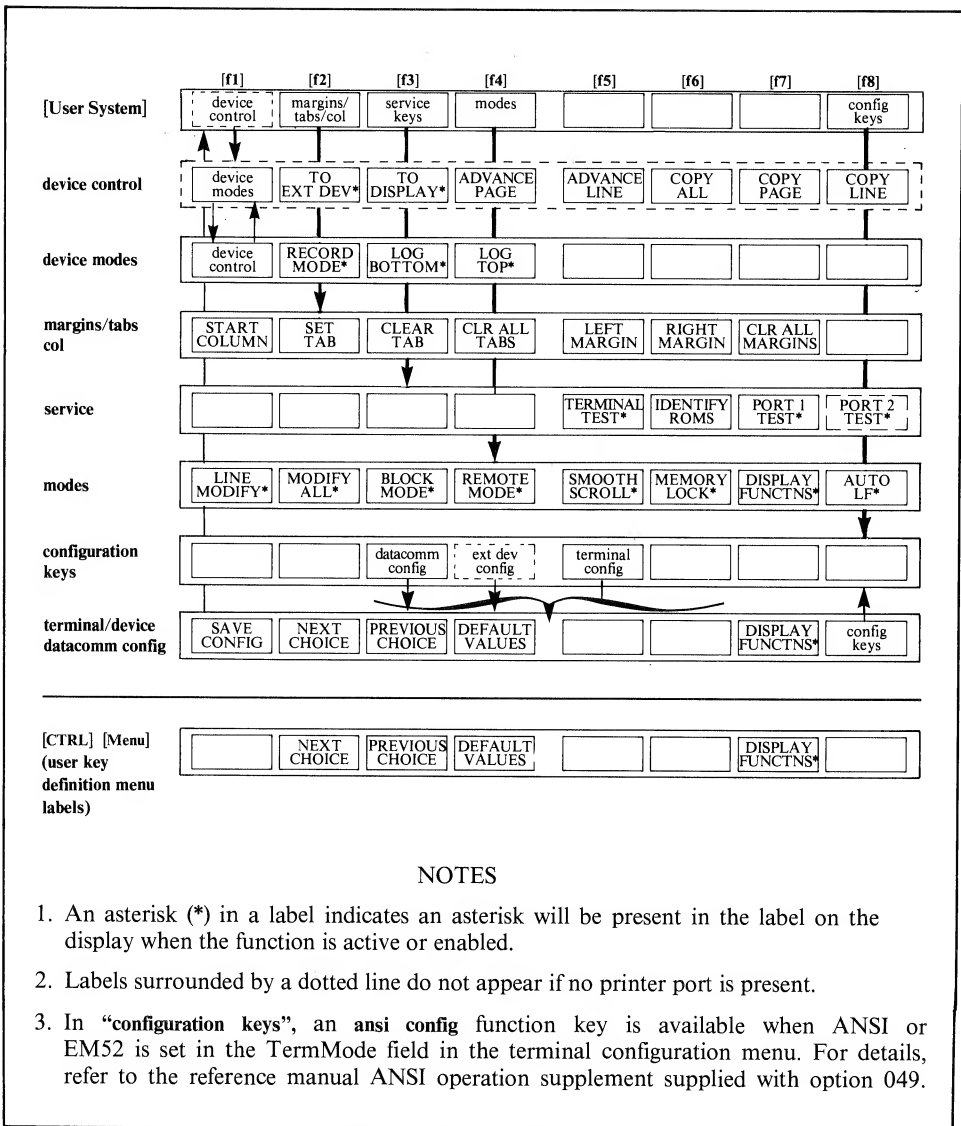


Figure 1-6. Function Key Hierarchy of the HP 2392A



HP 2394A Screen Label Tree

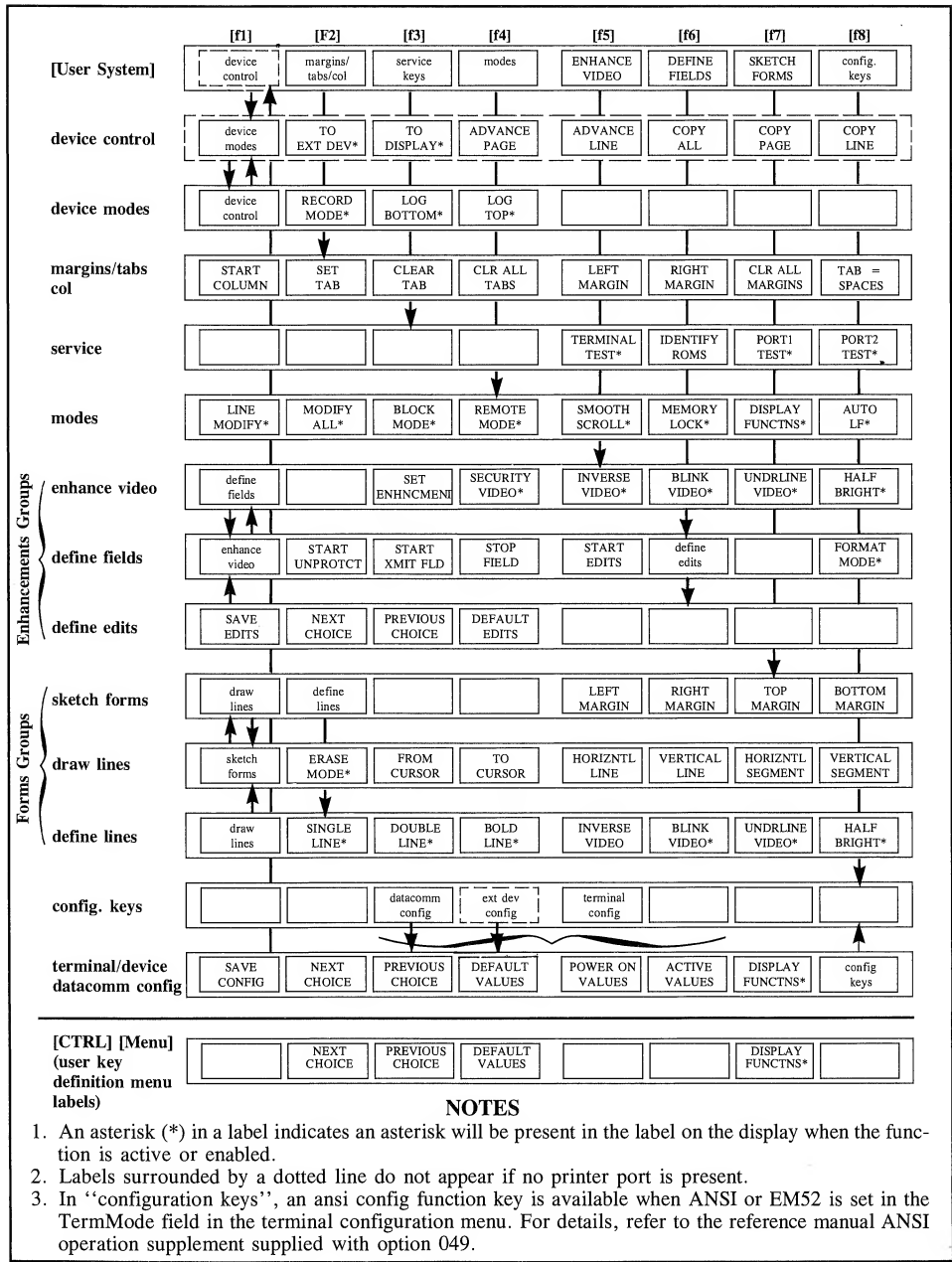


Figure 1-7. Function Key Hierarchy of the HP 2394A

When you press the **[CTRL]** and **[Menu]** keys simultaneously, the user keys definition menu (Figure 1-8) appears on the screen. By filling in this menu you can define the screen label and functional characteristics for eight user keys. Then to enable the eight user keys, press the **[Shift]** and **[User System]** keys (or **[CTRL]** and **[User System]**, or **[Menu]** twice). Figure 1-9 shows the default user key screen labels and Figure 1-10 shows some sample user-defined user key screen labels.

Pressing the **[Menu]** key removes the key labels from the screen. The user function keys, however, are still enabled. To re-enable the labels, press **[User System]** or **[Menu]** again.

Refer to Section III for a complete description of the function keys and information on defining these keys.

## STATUS LINE

This information is as follows:

- KB Lockd** – Keyboard locked out indicator. The keyboard is locked during a data transfer to the computer, when a handshake is pending, when the terminal is in record mode, and also by the program.
- \*** – Modem indicator (displayed when modem is operational). See Asterisk field in datacomm configuration menu, Section VI.
- ANSI or EM52** – Terminal mode indicator displayed when ANSI or EM52 set in the TermMode field in the terminal configuration menu (the selected mode is displayed in the middle of the status line, after the modem indicator). For details, refer to the reference manual ANSI operation supplement supplied with option 049.
- CAPS** – Keyboard set to capitals indicator (using **[Caps]** key).
- Ins Char** – Insert mode indicator (using **[Insert char]** key).
- STOP** – **[Stop]** key active indicator.

## CONFIGURING THE TERMINAL

The terminal contains no physical straps or switches (other than the OFF/ON switch on the front panel). You configure the terminal through the use of a configuration menu displayed on the screen.

Refer to Section II for complete information on configuring the terminal.

## TERMINAL PORTS

The standard terminal is equipped with one external port (Port 1), a combined RS232C/HP422 port used for the computer connection. The terminal may be optionally equipped with an interfacing module to provide a second port.

f1 E <sub>C</sub> p	T	LABEL	f1	
f2 E <sub>C</sub> q	T	LABEL	f2	
f3 E <sub>C</sub> r	T	LABEL	f3	
f4 E <sub>C</sub> s	T	LABEL	f4	
f5 E <sub>C</sub> t	T	LABEL	f5	
f6 E <sub>C</sub> u	T	LABEL	f6	
f7 E <sub>C</sub> v	T	LABEL	f7	
f8 E <sub>C</sub> w	T	LABEL	f8	

Figure 1-8. User Keys Definition Menu (default)

[f1]	[f2]	[f3]	[f4]	[f5]	[f6]	[f7]	[f8]
f1	f2	f3	f4	f5	f6	f7	f8

Figure 1-9. Default User Key Labels

[f1]	[f2]	[f3]	[f4]	[f5]	[f6]	[f7]	[f8]
LOG ON STRING	INITIATE NEWWORD	DELETE ORDER	UPDATE ORDER	DISPLAY ORDER	LIST ALL ORDERS	!HELP!	*EXIT*

Figure 1-10. Sample User-defined User Key Labels

KB Lockd			*	CAPS	Ins Char	STOP	

Figure 1-11. Screen Status Line

## COMPUTER PORT DATA COMMUNICATIONS

The terminal can operate at speeds ranging from 110 to 19200 baud.

Transmission can be performed in character mode, block line mode, or block page mode; in all cases the data may be either formatted (a data entry form with unprotected and protected fields) or unformatted.

Using the configuration process, you may enable the following forms of parity generation and checking :

- None
- Odd
- Even
- Ones (8th bit forced to 1)
- Zeros (8th bit forced to 0)

See Section VI for complete information.

## EXTERNAL PRINTER PORT (Optional)

The external port at the rear of the terminal (when fitted) provides interfacing for serial or parallel printers. The port may be configured to operate from 110 to 9600 baud; choice of 0's, 1's, odd, even, or no parity; 0 to 255 nulls; choice of XON/XOFF, SRR, or CS/CB handshaking.

See Section V for complete information.

## SELF-TEST

The terminal is engineered for high reliability, ease of testing, and rapid repair.

When the terminal's power is first turned on, a power-on self-test automatically verifies the integrity of all ROM (Read-Only Memory) and RAM (Random-Access Memory) chips within the terminal. The power-on self-test also does a verification of the configuration data stored in the non-volatile memory.

Using the system function keys, you may also initiate any of the following self-tests :

1. Port 1 or Port 2 test. This is a very flexible forms-selected test that verifies the integrity of the data communications ports. Loop-back via a test hood or a modem must be performed.
2. Terminal test. This self-test does a CRC (Cyclic Redundancy Check) verification of all ROM chips within the terminal, verifies the integrity of all RAM chips within the terminal, and then displays a test pattern on the screen. The test pattern includes all characters (and segments of the line drawing set) as well as all the character enhancements, plus the terminal status (see Section VIII).
3. Identify ROMs. This self-test generates a listing (on the display screen) of all installed ROMs.

See Section VIII for complete information concerning the various self-tests.

# Configuring the Terminal

SECTION

II

## INTRODUCTION

The terminal is designed so that the various terminal characteristics can be configured quickly and easily by displaying configuration “menus” on the screen and then using function keys to change the content of these menus. Thus altered, the terminal’s configuration characteristics are stored in non-volatile memory.

The content of these menus may also be altered from a program (executing in a host computer) through the use of escape sequences. The changes made by the host computer are temporary and will be lost through hard reset or power down. (In the HP 2394A terminal, you can make permanent changes to the menus. (See page 2-16.)

## CONFIGURATION FUNCTION KEYS

To gain access to the configuration menus through the keyboard, press the [User System] key. This causes the following label display across the bottom of the screen (where [f1], [f2], etc., refer to the function key corresponding to the label):

	[f1]	[f2]	[f3]	[f4]	[f5]	[f6]	[f7]	[f8]
HP 2392A:	device control	margin/ tab/col	service keys	modes				config keys
HP 2394A:	device control	margins/ tab/col	service keys	modes	enhance video	define fields	sketch forms	config keys

### NOTE

The “device control” label is present only on terminals equipped with Port 2.

Pressing **config keys** ([f8]) changes the label display to

[f1]	[f2]	[f3]	[f4]	[f5]	[f6]	[f7]	[f8]
		datacomm config	ext dev config	terminal config			

### NOTE

The “ext dev config” label is present only on terminals equipped with Port 2. An “ansi config” label is present when ANSI or EM52 is selected in the terminal configuration menu (for details, refer to the reference manual ANSI supplement supplied with option 049).

Pressing **datacomm config** ([f3]), **ext dev config** ([f4]) or **terminal config** ([f5]) cause a configuration “menu” to appear on the screen and redefines the function keys to a set of functions that will assist you in manipulating the various parameters within the menu.

The datacomm config menu is described in Section VI, Data Communications. The ext dev config menu is described in Section V, Printer Control. The terminal configuration menu is described in this section. In addition, information is provided which will enable you to change the terminal configuration menu from the keyboard.

## TERMINAL CONFIGURATION MENU

When you press the **terminal config** ([f5]) function key, the menu and function key display shown in Figure 2-1 (HP 2392A) or Figure 2-2 (HP 2394A) appear on the screen. Note that the menu shown in Figure 2-1 (HP 2392A) or Figure 2-2 (HP 2394A) contains the default settings for all the fields. If you had previously changed the content of any of the fields and then saved the menu in non-volatile memory, the menu would appear on the screen as you had configured it.

### NOTE

Whenever a configuration menu is on the screen, incoming data, including escape sequences, is stored in buffer (up to 256 bytes) and not processed and no data is transmitted, until the menu is exited. (Refer to Section VI for pacing considerations).

The menu contains a set of unprotected fields that you access using the [TAB▶◀] key. Except when the cursor is positioned in the fields labeled "Terminal Id", "Return Def", "Fld Separator", or "Blk Terminator", the alphanumeric keys are disabled and you select the desired parameters using the NEXT CHOICE ([f2]) and PREVIOUS CHOICE ([f3]) function keys.

The meanings of the various fields are described in Table 2-1. The fields are listed in the order: terminal ID, language, keyboard function, forms mode (HP 2394A only) datacomm functions and external device functions.

TERMINAL CONFIGURATION							
Datacomm/ExtDev	PORT1/PORT2		Keyboard	USASCII			
TerminalId	2392A		Language	ENGLISH			
LocalEcho	OFF	CapsLock	OFF	Start Col	01	Bell	ON
XmitFunctn (A)	NO	SPOW (B)	NO	InhEolWrp (C)	NO	Line/Page (D)	LINE
InhHndShk (G)	NO	Inh DC2 (H)	NO	EscXfer (N)	NO		
FldSeparator	U <sub>S</sub>	BlkTerminator	R <sub>S</sub>	ReturnDef	C <sub>R</sub>	TermMode	HP
<div style="display: flex; justify-content: space-around; margin-top: 10px;"> <div style="border: 1px solid black; padding: 2px; text-align: center;">[f1] SAVE CONFIG</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">[f2] NEXT CHOICE</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">[f3] PREVIOUS CHOICE</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">[f4] DEFAULT VALUES</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">[f5]</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">[f6]</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">[f7] DISPLAY FUNCTNS</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">[f8] config keys</div> </div>							

Note: The "Datacomm/Ext dev" field is only displayed when the terminal is equipped with a second port.  
The "Caps Lock" field only affects USASCII keyboard (ignored by terminal with national keyboard).

Figure 2-1. Terminal Configuration Menu of the HP 2392A

TERMINAL CONFIGURATION							
Datacomm/ExtDev		<input type="text" value="PORT1/PORT2"/>		Keyboard		<input type="text" value="USASCII"/>	
Terminal ID		<input type="text" value="2394A"/>		Language		<input type="text" value="ENGLISH"/>	
Local Echo	<input type="text" value="OFF"/>	CapsLock	<input type="text" value="OFF"/>	Start Col	<input type="text" value="01"/>	Bell	<input type="text" value="NO"/>
XmitFnctn (A)	<input type="text" value="NO"/>	SPOW (B)	<input type="text" value="NO"/>	InhEolWrp (C)	<input type="text" value="NO"/>	Line/Page (D)	<input type="text" value="LINE"/>
InhHndShk (G)	<input type="text" value="NO"/>	Inh DC2 (H)	<input type="text" value="NO"/>			Esc Xfer (N)	<input type="text" value="NO"/>
						FormsBufSiz (256x)	<input type="text" value="000"/>
FldSeparator	<input type="text" value="us"/>	BlkTerminator	<input type="text" value="rs"/>	Return=Enter	<input type="text" value="NO"/>	ReturnDef	<input type="text" value="CR"/>
Tab=Spaces	<input type="text" value="NO"/>					TermMode	<input type="text" value="HP"/>
FORMAT MODE							
Decimal Type	<input type="text" value="US"/>	Imp Dec Digits	<input type="text" value="2"/>	Transmit	<input type="text" value="All Fields"/>	Print	<input type="text" value="Fields"/>
<b>[f1]</b>	<b>[f2]</b>	<b>[f3]</b>	<b>[f4]</b>	<b>[f5]</b>	<b>[f6]</b>	<b>[f7]</b>	<b>[f8]</b>
SAVE CONFIG	NEXT CHOICE	PREVIOUS CHOICE	DEFAULT VALUES	POWER ON VALUES	ACTIVE VALUES	DISPLAY FUNCTIONS	config keys

Note: The "Datacomm/Ext dev" fields is only displayed when the terminal is equipped with a second port.  
The "Caps Lock" field only affects USASCII keyboard (ignored by terminal with national keyboard).

Figure 2-2. Terminal Configuration Menu of the HP 2394A

Table 2-1. Terminal Configuration Menu Fields

<b>Terminal Id</b>	<p>This field allows the user to specify which Terminal Identification will be sent to the host computer upon a Terminal Id request (E<sub>C</sub> * s ^)</p> <p>Values: Any string of up to five characters.</p> <p>Default: 2392A for the HP 2392A and 2394A for the HP 2394A</p> <p>Only the default values (2392A/2394A) are supported by Hewlett-Packard.</p>
<b>Keyboard</b>	<p>The terminal can be equipped with one of several keyboards (see table 1-1). This field allows the keyboard type to be selected, as the position of keys generating the same character differs from keyboard to keyboard, and the keycap labels are different (according to the country).</p> <p>The terminal includes as standard part of the ROMAN 8 Character Set, which includes the USASCII set plus an extended character set that supports the special characters associated with Western European languages (see Appendix B).</p> <p>Values: USASCII (United States), DANSK (Danish), NEDERLANDS (Dutch), SUOMI (Finnish), FRANÇAIS (French), ENGLISH CANADIAN, CANADIEN FRANÇAIS (French Can.), DEUTSCH (German), VLAAMS (Flemish), ITALIANA (Italian), NORSK (Norwegian), ESPAÑOL EUR. (European Spanish), ESPAÑOL LAT. (Latin American Spanish), SVENSK (Swedish), SCHWEIZ-DEUTSCH (Swiss German), SUISSE ROMAND (Swiss French), UK (United Kingdom).</p> <p>Default: USASCII</p>
<b>Language</b>	<p>This field specifies in which language all function key labels and messages will be displayed.</p> <p>You may configure the terminal so that the value given to the language used for function keys and messages is different from the value given to the keyboard field. For example, a French programmer may use a USASCII keyboard and still have the function keys labels in French, in that case, all characters displayed, received or transmitted will be USASCII characters.</p> <p>Values: ENGLISH  FRANÇAIS (French)  DEUTSCH (German)  ITALIANA (Italian)  ESPAÑOL (Spanish)  NEDERLANDS (Dutch)  NORSK (Norwegian)  SVENSK (Swedish)  SUOMI (Finnish)  DANSK (Danish)</p> <p>Default: ENGLISH</p>

Continued . . .



Table 2-1. Terminal Configuration Menu Fields (Continued)

<b>Bell</b>	<p>This specifies whether the terminal's bell is enabled or disabled. When disabled, the bell will not sound when the cursor nears the end of line nor when the cursor advances from one field to the next in a formatted display. The bell will, however, still sound in response to an ASCII Bell control code (decimal 7; control G).</p> <p>Values: ON (Bell enabled, <math>E_C</math> &amp; k 1 D) OFF (Bell disabled, <math>E_C</math> &amp; k 0 D)</p> <p>Default: ON</p>
<b>Caps Lock</b>	<p>This field specifies whether the terminal generates the full 128-character ASCII set or only Teletype-compatible codes.</p> <p>ON (<math>E_C</math> &amp; k 1 C) The terminal generates only Teletype-compatible codes: uppercase ASCII (00-5F, hex) and DEL (7F, hex). Unshifted alphabetic keys (a-z) generate the codes for their uppercase equivalents. The {,  , and } keys generate the codes for [, \, and ], respectively. The key for generating ~ and ` is disabled.</p> <p>OFF (<math>E_C</math> &amp; k 0 C) The terminal generates the full 128-character ASCII set of codes.</p> <p>Note that the Caps Lock field only affects USASCII keyboards, the field is ignored when a national keyboard is specified in the keyboard field.</p> <p>Default: OFF</p>
<b>SPOW (B)</b>	<p>This field specifies whether or not spaces entered through the keyboard will overwrite existing characters.</p> <p>NO (<math>E_C</math> &amp; s 0 B) Spaces entered through the keyboard will overwrite existing characters.</p> <p>YES (<math>E_C</math> &amp; s 1 B) Enable Space OverWrite (SPOW) latch. Once enabled, the SPOW latch is turned on by a carriage return and is turned off by a line feed, home up, or tab. It may also be turned on and off programmatically through the use of an <math>E_C</math> &amp; k sequence as follows:</p> <p>ON (<math>E_C</math> &amp; k 1 N) OFF (<math>E_C</math> &amp; k 0 N)</p> <p>When the SPOW latch is off, overwriting occurs as normal. When the SPOW latch is on, spaces entered through the keyboard move the cursor forward but do not overwrite existing characters.</p> <p>When a configuration or user key menu is displayed, the SPOW (B) field is ignored (equivalent to "NO").</p> <p>Default: NO</p>

Continued . . .

Table 2-1. Terminal Configuration Menu Fields (Continued)

<b>InhEolWrp (C)</b>	<p>This field specifies whether or not the end-of-line wrap is inhibited.</p> <p>NO (<math>E_C \&amp; s0C</math>) When the cursor reaches the right margin it automatically moves to the left margin in the next lower line (a local carriage return and line feed are generated).</p> <p>YES (<math>E_C \&amp; s1C</math>) When the cursor reaches the right margin it remains in that screen column until an explicit carriage return or other cursor movement function is performed (succeeding characters overwrite the existing character in that screen column).</p> <p>Default: NO</p>
<b>Return=Enter (HP 2394A)</b>	<p>This field specifies whether or not the [RETURN] key is to function as though it were the [ENTER] key. The value of "YES" causes both keys to function in the manner defined for the [ENTER] key when the terminal is in remote mode. The value "NO" causes each key to function according to its own definition.</p> <p>NO (<math>E_C \&amp; q8te1\{OR</math>) YES (<math>E_C \&amp; q8te1\{1R</math>)</p> <p>Default: NO</p>
<b>Tab=Spaces (HP 2394A)</b>	<p>When this feature is enabled, pressing the [TAB] key generates the number of ASCII space codes required to move the cursor forward to the next tab stop. If no tab stops exist between the current cursor position and the end of the line, the bell sounds and no spaces are generated. Similarly, pressing the [SHIFT] [TAB] key generates the number of ASCII backspace codes required to move the cursor backward to the preceding tab stop.</p> <p>NO (<math>E_C \&amp; q8te1\{0T</math>) YES (<math>E_C \&amp; q8te1\{1T</math>)</p> <p>Default: NO</p>
<b>Forms Buf Size (256x) (HP 2394A)</b>	<p>This terminal configuration field is used to select the amount of terminal memory (RAM) allocated to forms cache. Memory is allocated in blocks of 256 bytes. In the config menu, the field FormsBufSize (256x) is used to specify the number of blocks allocated to cache memory. Although the value of this field can range from 0 to 255, the maximum amount of cache that can be allocated is about 125 blocks (32k bytes). Any request to assign more memory than is physically available will be reduced to the upper limit (see section 4 for more information on forms cache operation). Default: m=4 x=0 The variable "m" identifies the terminal configuration menu. The value of "m" may be a number from 4-7. Decimal integer "x" within range 0-255.</p> <p>(<math>E_C \&amp; q &lt; m &gt; te 2 \{ &lt; x &gt; L</math>)</p>
<b>Implied Decimals (HP 2394A)</b>	<p>This configuration field is used in conjunction with the local edit feature. It is used to specify the number of places to the right of the decimal in an implied decimal field. The value of Implied Dec Digits may range from 0 to 9.</p> <p>(<math>E_C \&amp; k &lt; x &gt; Y</math>) Default: x=2</p>

Table 2-1. Terminal Configuration Menu Fields (Continued)

<b>Decimal Type</b> <b>(HP 2394A)</b>	<p>This configuration parameter is used by the local edit feature when processing decimal, or implied decimal fields. It is used to specify whether the decimal point is to be in U.S. (.) or European (,) notation. The valid values for this config field are: US and EUR.</p> <p>US (E<sub>C</sub>&amp;k 0x)  EUR (E<sub>C</sub>&amp;k 1x)  Default: US</p>
<b>Transmit</b> <b>(HP 2394A)</b>	<p>This field specifies whether you want all fields or only those fields which have been modified to be transmitted from a formatted display.</p> <p>Values: &lt;All Fields&gt; (E<sub>C</sub>&amp;k 0Z)  &lt;Modified Fields&gt; (E<sub>C</sub>&amp;k 1Z)</p> <p>Default: &lt;All Fields&gt;</p>
<b>Print</b> <b>(HP 2394A)</b>	<p>This field controls printer output. When in format mode a &lt;COPY&gt; command will either print only the unprotected fields from display memory or all of display memory, depending on the selected field.</p> <p>&lt;Fields&gt;: Prints only the unprotected fields from display memory.</p> <p>&lt;ALL&gt;: Prints all of display memory</p> <p>Default: &lt;FIELDS&gt;</p>
<b>Datacomm/Ext Dev</b>	<p>This field is present only when the terminal is equipped with two ports. It specifies which port (Port 1 or Port 2) is assigned to the host computer and which is assigned to the external printer.</p> <p>Values: Port 1/Port 2: Port 1 host computer, Port 2 printer.  Port 1/Port 2: Port 1 printer, Port 2 host computer.</p> <p>Default: Port 1/Port 2</p>
<b>LocalEcho</b>	<p style="text-align: center;">NOTE</p> <p>Attempting to assign a parallel datacomm (printer) port to the host computer causes the error message "Function locked Press [RETURN] to clear" to be displayed.</p> <p>This field specifies whether characters entered through the keyboard are both displayed on the screen and transmitted to the host computer.</p> <p>ON (E<sub>C</sub>&amp;k 1L)  Characters entered through the keyboard are both displayed on the screen and transmitted to the host computer.</p> <p>OFF (E<sub>C</sub>&amp;k 0L)  Characters entered through the keyboard are transmitted to the host computer only (if they are to appear on the screen, the host computer must "echo" them back to the terminal.)</p> <p>Default: OFF</p>

Table 2-1. Terminal Configuration Menu Fields (Continued)

<b>Start Col</b>	<p>If the line in which you are entering data is the bottom most used line in display memory (there are no printing or non-printing characters following the current line in display memory), the terminal automatically generates a logical start-of-text pointer to designate the leftmost character that you enter in the line.. This pointer remains with the line in display memory until the line is deleted.</p> <p>When you are operating in MODIFY LINE or MODIFY ALL mode and you press [Enter] or [Return], the data transmission from the terminal normally begins at the logical start-of-text pointer in the particular line. If the line has no logical start-of-text pointer, however, the data transmission begins at the designated start column. This designated start column can be defined and saved in non-volatile memory using the <b>StartCol</b> field of the terminal configuration menu. The active value of this field can also be temporarily redefined using one of the “margin/tab/col” function keys.</p> <p>Values: 1 – 80</p> <p>Default: 1</p>
<b>XmitFunctn(A)</b>	<p>This field specifies whether escape code functions are both executed at the terminal and transmitted to the host computer.</p> <p>YES (<math>\text{E}_C</math> &amp; s 1A)  The escape code sequences generated by control keys such as [Insert char] and [Delete char] are transmitted to the host computer. If local echo is ON, the function is also performed locally.</p> <p>NO (<math>\text{E}_C</math> &amp; s 0A)  The escape code sequences for the major function keys are executed locally but NOT transmitted to the host computer.</p> <p>Note that with Xmit Functn (A) enabled, pressing the “display functions” function key will emit <math>\text{E}_C</math> Y to a host computer (<math>\text{E}_C</math> Z is never transmitted).</p> <p>Default: NO</p>
<b>InhHndShk (G)</b> and <b>Inh DC2 (H)</b> (see Section VI)	<p>Together, these fields determine what type of handshaking is to be used when transferring blocks of data from the terminal to the host computer.</p> <p>The various types of block transfers that may occur are as follows:</p> <ul style="list-style-type: none"> <li>• A data transfer initiated by pressing the [Enter] key in character, block line, or block page mode.</li> <li>• A data transfer initiated by pressing the [Enter] or [Return] key in modify mode.</li> <li>• A data transfer initiated by pressing a transmit only (T) user key ([f1] through [f8]), or the [Select] key.</li> </ul>

Continued...

Table 2-1. Terminal Configuration Menu Fields (Continued)

	<ul style="list-style-type: none"> <li>• The terminal's response to a cursor sense, terminal ID status, primary status, secondary status, or device status request issued from the host computer.</li> <li>• The device control completion code (S, F or U) transmitted by the terminal in conjunction with a device control operation initiated by the host computer.</li> </ul> <p>When performing block transfers, there are three possible handshakes:</p> <ol style="list-style-type: none"> <li>1. No handshake; terminal merely transmits block of data.</li> <li>2. Computer sends <math>D_1</math>; terminal transmits block of data (<math>D_1</math> handshake).</li> <li>3. Computer sends <math>D_1</math>; terminal responds with <math>D_2</math>; computer responds with another <math>D_1</math>; terminal transmits block of data (<math>D_1/D_2/D_1</math> handshake).</li> </ol> <p style="text-align: center;">NOTE</p> <p>When the <math>D_1/D_2/D_1</math> handshake is enabled and the <b>line/page</b> field of terminal config is selected to be "line", a <math>C_R</math> or <math>C_R L_F</math> is transmitted after the <math>D_2</math>. If "line" is not selected, nothing is transmitted after <math>D_2</math>.</p> <p>In general, the InhHndShk (G) and Inh DC2 (H) fields have the following effects:</p> <table> <tr> <td>InhHndShk (G) = YES</td><td>Eliminates the use of the <math>D_1</math> handshake (terminal will either use the <math>D_1/D_2/D_1</math> handshake or no handshake at all).</td></tr> <tr> <td>Inh DC2 (H) = YES</td><td>Eliminates the use of the <math>D_1/D_2/D_1</math> handshake (terminal will either use the <math>D_1</math> handshake or no handshake at all).</td></tr> <tr> <td>Both = YES</td><td>No handshake.</td></tr> </table> <p>Specifically, however, the type of handshaking used for block transfers is determined by a combination of the following factors:</p> <ol style="list-style-type: none"> <li>1. The type of block transfer to be performed.</li> <li>2. What mode the terminal is currently operating in (character, block line, block page, or modify mode).</li> <li>3. The setting of the InhHndShk (G) and Inh DC2 (H) fields.</li> </ol>	InhHndShk (G) = YES	Eliminates the use of the $D_1$ handshake (terminal will either use the $D_1/D_2/D_1$ handshake or no handshake at all).	Inh DC2 (H) = YES	Eliminates the use of the $D_1/D_2/D_1$ handshake (terminal will either use the $D_1$ handshake or no handshake at all).	Both = YES	No handshake.
InhHndShk (G) = YES	Eliminates the use of the $D_1$ handshake (terminal will either use the $D_1/D_2/D_1$ handshake or no handshake at all).						
Inh DC2 (H) = YES	Eliminates the use of the $D_1/D_2/D_1$ handshake (terminal will either use the $D_1$ handshake or no handshake at all).						
Both = YES	No handshake.						

Continued . . .

Table 2-1. Terminal Configuration Menu Fields (Continued)

If your terminal is connected to a Hewlett-Packard computer system, you will find that the default settings for these fields (both NO) are usually adequate for your purposes. If you are concerned about the specific type of handshake to be used for one or more of the particular types of block transfers, however, you should use the following summary to verify (or alter) the settings of the InhHndShk (G) and Inh DC2 (H) fields (also see Section VI):

- Transmit only (T) user key or [Select] key in block line or character mode; or

Cursor sense, terminal ID status, primary status, secondary status, display transfer initiated by  $\epsilon_C$  d or device status request; or

Device control completion code:

InhHndShk (G) = NO	}	$\rightarrow D_1$
InhHndShk (H) (ignored)		
InhHndShk (G) = YES	}	$\rightarrow D_1/D_2/D_1$
Inh DC2 (H) = NO		
InhHndShk (G) = YES	}	$\rightarrow$ no handshake
Inh DC2 (H) = YES		

- [Enter] key in block mode; or

Transmit only (T) user key or [Select] key in block page mode:

InhHndShk (G) (ignored)	
Inh DC2 (H) = NO	$\rightarrow D_1/D_2/D_1$
Inh DC2 (H) = YES	$\rightarrow$ no handshake

- [Enter] key in character mode; or

[Enter] or [Return] key in modify mode:

InhHndShk (G) = YES	}	$\rightarrow D_1/D_2/D_1$
Inh DC2 (H) = NO		

Any other combination  $\rightarrow$  no handshake

Continued . . .

Table 2-1. Terminal Configuration Menu Fields (Continued)

	<p>The G and H selections can be temporarily selected by the following escape sequences:</p> <p> <math>\text{E}_C \&amp; s 0G</math>    disables G  <math>\text{E}_C \&amp; s 0H</math>    disables H  <math>\text{E}_C \&amp; s 1G</math>    enables G  <math>\text{E}_C \&amp; s 1H</math>    enables H </p> <p>Defaults: InhHndShk (G) = NO  Inh DC2 (H) = NO</p>
<b>Line/Page (D)</b>	<p>This field specifies whether or not the terminal when operating in block mode, will transmit data a line at a time or a page at a time.</p> <p><b>Line (<math>\text{E}_C \&amp; s 0D</math>)</b>  When operating in block mode, the terminal will transmit data a line at a time.</p> <p><b>Page (<math>\text{E}_C \&amp; s 1D</math>)</b>  When operating in block mode, the terminal will transmit the data in the display memory, either from the beginning of the memory or the current cursor position (see the Enter key, Section III).</p> <p>For detailed description of the differences between block line and block page mode, refer to "ENTER KEY" in Section III of this manual.</p> <p>Default: LINE</p>
<b>FldSeparator</b>	<p>When you press the [Enter] key while the terminal is in block page mode and display memory contains a formatted display, the terminal automatically transmits the specified field separator character at the end of each protected field (except the final one).</p> <p>Value: Any ASCII character (or ROMAN 8 character in 8-bit operations)</p> <p>Default: <math>\text{U}_5</math></p>
<b>BlkTerminator</b>	<p>For data transfers between the terminal and a host computer, the terminal (under certain circumstances) transmits the specified block terminator character at the end of the transfer operation. For details, see "The ENTER Key", in Section III.</p> <p>This character, when encountered in display memory, terminates a data transfer ([Enter] key transmissions).</p> <p>Value: Any ASCII character</p> <p>Default: <math>\text{R}_5</math></p>

Continued . . .

Table 2-1. Terminal Configuration Menu Fields

<b>RETURN def</b>	<p>This field specifies the definition of the <b>[Return]</b> key. The default definition is an ASCII<sub>R</sub>. The definition may consist of up to two characters. If the second character is a space, it is ignored.</p> <p>Default: <math>c_R</math></p>
<b>Esc Xfer (N)</b>	<p>This field controls the transfer of escape sequences to a printer, and has no effect if there is no second port.</p> <p><b>YES</b> (<math>E_c \&amp;s 1N</math>) = When transferring data from display memory to an external printer, each line automatically starts with <math>s_1 E_c \&amp;d @</math> (this is to return to the primary character set and to stop any character enhancements). In addition, escape sequences relating to the display (such as those specifying display enhancements, format mode fields, and alternate character sets) are sent to the external printer if encountered within the data.</p> <p><b>NO</b> (<math>E_c \&amp;s 0N</math>) = Escape sequences relating to the display are not sent to the external printer (and <math>s_1 E_c \&amp;d @</math> is not sent).</p> <p><b>NOTE:</b> The Esc Xfer (N) field only affects data transfers between display memory and an external printer. It does NOT affect <math>\langle Esc \rangle \&amp;pW</math> data transfers that go directly from the host computer to the external computer.</p> <p>Default: NO</p>
<b>TERM MODE</b>	<p>Values: HP ANSI (<math>E_c \&amp;k1 \backslash</math> = change from HP to ANSI) EM52</p> <p>Default: HP</p> <p>When the default mode (HP) is specified, the terminal operates as described in this manual. When ANSI or EM52 is specified, the terminal should be operated and programmed as described in the reference manual ANSI operation supplement (supplied when option 049 is ordered). In addition, the selected mode is displayed in the status line and an ANSI configuration menu is available.</p>

Note that as you alter the fields of a configuration menu on the screen, the selected values do NOT alter the content of non-volatile memory nor do they have any effect on the operation of the terminal.

When you have set all the fields to the desired values, you may then save them in non-volatile memory using the **SAVE CONFIG** (**[f1]**) function key.

When you do this, the chosen values take effect immediately, and the terminal returns to normal operations with the **SYSTEM** labels displayed.

While the terminal configuration menu is displayed on the screen, the **[f4]**, **[f7]**, and **[f8]** function keys perform as shown in Table 2-2.



Table 2-2. Configuration Function Keys [f4], [f7], [f8]

<p>[f4]</p> <p>DEFAULT VALUES</p>	<p>Pressing this key causes all fields in the menu on the screen to be filled with their default values. (For default values, see figure 2-1).</p>
<p>[f7]</p> <p>DISPLAY FUNCTNS</p>	<p>Pressing this key alternately enables and disables display functions mode. When enabled, an asterisk appears in the function key display. You use display functions mode for entering control characters in the <b>Return Def</b>, <b>Fld Separator</b>, or <b>Blk Terminator</b> fields. Control characters may be accessed using [CTRL] and the relevant alphabetic key (see Appendix B). Note that this implementation of display functions mode is separate from that which is enabled/disabled via the mode selection keys. Enabling or disabling display functions mode using this function key does NOT alter the effect of the <b>DISPLAY FUNCTNS</b> mode selection key (and vice versa).</p>
<p>[f8]</p> <p>config keys</p>	<p>Pressing this key removes the menu from the screen (WITHOUT activating it or saving it in non-volatile memory) and returns the function key labels to the following:</p> <div data-bbox="219 857 1069 927"> <div>[f1]</div> <div>[f2]</div> <div>[f3] datacomm config</div> <div>[f4] ext dev config</div> <div>[f5] terminal config</div> <div>[f6]</div> <div>[f7]</div> <div>[f8]</div> </div> <p>The “ext dev config” label is blanked if Datacomm Module not installed (Port 2)</p>

## Lock/Unlock Configuration Menus

Using an escape sequence, you can “lock” the current configuration menus (terminal config or datacomm config) so that the menu can not be accessed from the keyboard. Any attempt to access a locked menu from the keyboard will result in a “beep” from the bell and the “Function locked Press RETURN to clear” error message to be displayed. Note that when the configuration menus are locked, the **MODIFY ALL** ([f2]), **BLOCK MODE** ([f3]), **REMOTE MODE** ([f4]), **AUTO LF** ([f8]) mode selection keys are also locked.

To lock the menus, use the following escape sequence:

$\text{ESC} \& \text{q 1L}$

To unlock the menus, use the following escape sequence:

$\text{ESC} \& \text{q 0L}$

This is the only escape sequence to be retained in non-volatile memory on the HP 2392A terminal.

## Setting Terminal Configuration Parameters With Escape Codes

To set the terminal configuration parameters using escape codes, you must use an  $\text{E}_C\&k$  or  $\text{E}_C\&s$  sequences, depending upon which parameters you wish to set.

Parameter Name As Shown in Menu	Type of Escape Sequence Used
Local Echo	$\text{E}_C\&k$
Caps Lock	
Bell	
Spow Latch	
Decimal Type (HP 2394A)	
Implied Decimal (HP 2394A)	
Transmit (HP 2394A)	$\text{E}_C\&s$
Xmit Fnctn (A)	
SPOW (B) (enables SPOW latch)	
InhEolWrp (C)	
Line/Page (D)	
InhHndShk (G)	
Inh DC2 (H)	
Esc Xfer (N)	

The  $\text{E}_C\&k$  and  $\text{E}_C\&s$  sequences alter the particular parameter in the menu, and the new setting takes effect immediately, but they do NOT alter the content of non-volatile memory.

To change the active value of the Local Echo, Caps Lock or SPOW parameters, use an escape sequence of the following form:

Local Echo = OFF:	$\text{E}_C\&k\ 0L$
Local Echo = ON:	$\text{E}_C\&k\ 1L$
Caps Lock = OFF:	$\text{E}_C\&k\ 0C$
Caps Lock = ON:	$\text{E}_C\&k\ 1C$
Bell = OFF:	$\text{E}_C\&k\ 0D$
Bell = ON:	$\text{E}_C\&k\ 1D$
Spow Latch = OFF:	$\text{E}_C\&k\ 0N$
Spow Latch = ON:	$\text{E}_C\&k\ 1N$
Decimal Type = US:	$\text{E}_C\&k\ 0X$
Decimal Type = EUR:	$\text{E}_C\&k\ 1X$
Implied Decimals = 0-9	$\text{E}_C\&k\ <x>Y$
Transmit = All Fields	$\text{E}_C\&k\ 0Z$
Transmit = Modified Fields	$\text{E}_C\&k\ 1Z$

You may combine these and other  $E_C$  &  $k$  parameters within one escape sequence. If you do, the final identifier (C or L or N) must be uppercase, and all preceding identifiers must be lowercase. For example, to set **Local Echo=ON** and **Caps Lock=ON**, you could use either of the following escape sequences:

$$E_C \& k 11 1C$$

$$E_C \& k 1c 1L$$

To change the active values of any of the following parameters, use an escape sequence of the form:

**Xmit Fnctn (A) = NO:**  $E_C \& s 0A$

**Xmit Fnctn (A) = YES:**  $E_C \& s 1A$

**Spow (B) = NO:**  $E_C \& s 0B$

**Spow (B) = YES:**  $E_C \& s 1B$

**InhEolWrp (C) = NO:**  $E_C \& s 0C$

**InhEolWrp (C) = YES:**  $E_C \& s 1C$

**Line/Page (D) = LINE:**  $E_C \& s 0D$

**Line/Page (D) = PAGE:**  $E_C \& s 1D$

**InhHndShk (G) = NO:**  $E_C \& s 0G$

**InhHndShk (G) = YES:**  $E_C \& s 1G$

**Inh DC2 (H) = NO:**  $E_C \& s 0H$

**Inh DC2 (H) = YES:**  $E_C \& s 1H$

**Esc Xfer (N) = NO:**  $E_C \& s 0N$

**Esc Xfer (N) = YES:**  $E_C \& s 1N$

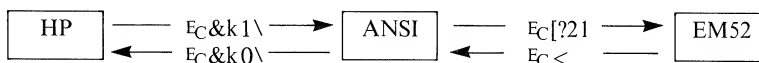
You may combine these and other  $E_C$  &  $s$  parameters in one escape sequence. If you do, the final identifier (A or B or C or D or G or H or N) must be uppercase and all preceding identifiers must be lowercase. For example, to set **Line/Page (D) = PAGE**, **InhHndShk (G) = NO**, and **Inh DC2 (H) = YES**, you could use any of the following escape sequences:

$$E_C \& s 1d 0g 1H$$

$$E_C \& s 0g 1h 1D$$

$$E_C \& s 1h 1d 0G$$

Note that the terminal mode can be configured programmatically but only in the following directions:



Refer to Section X for details on ANSI and EM52 modes.

## Terminal Configuration Parameters Unique To The HP 2394A

The HP2394A terminal allows the user to modify the permanent configuration via escape sequences. Modes and strap settings can also be modified. This section explains and lists these escape sequences. To set the terminal configuration parameters programmatically, you must use an  $\text{E}_C$  &k ,  $\text{E}_C$  &s, or  $\text{E}_C$  &q sequence, depending upon which parameters you wish to set and whether or not you wish to alter the content of non-volatile memory.

The  $\text{E}_C$  &k and  $\text{E}_C$  &s sequences alter the particular parameter in the menu, and the new setting takes effect immediately, but it does not alter the content of non-volatile memory.

The  $\text{E}_C$  &q sequence, on the other hand, alters the particular parameter in non-volatile memory. The new configuration values become active immediately.  $\text{E}_C$  &q always overwrites the active values as well as those in non-volatile memory. If the active values have been changed prior to the  $\text{E}_C$  &q sequence (via  $\text{E}_C$  &k or  $\text{E}_C$  &s), those changes to the active values will be lost and may need to be made again.

When you issue an  $\text{E}_C$  &q sequence the terminal normally takes the current menu values from non-volatile memory and then alters only those fields (parameters) that you specifically include in the escape sequence. If you include the command parameter "d" at the start of the escape sequence, however, the terminal will start with the default values and then alter only those fields (parameters) that you specifically include in the escape sequence.

### Example: Configuration Modification

1. Press [User System], then <config keys> [f8], followed by <terminal config> [f5].
2. Press <DEFAULT VALUES> [f4], then <SAVE CONFIG> [f1].
3. Type in  $\text{E}_C$  &k1C. This will turn the Caps Lock field on.
4. Now turn the terminal off and then back on. Repeat Step 1. Notice that the Caps Lock field now reads OFF. The non-volatile memory was not altered when we used the  $\text{E}_C$  &k escape sequence.
5. Now let's use the  $\text{E}_C$  &q escape sequence. Type in  $\text{E}_C$  &q4te {1C.
6. Now turn the terminal off and then back on. Repeat Step 1. Now the Caps Lock field reads ON. Therefore, the non-volatile memory was altered when we used the  $\text{E}_C$  &q escape sequence.
7. To return to the default values, repeat Step 2.

Care should be used in setting default values for the terminal configuration programmatically. In addition to defaulting the terminal portion of the terminal configuration, the Remote Mode, Block Mode, Modify All, and AutoLF modes are also defaulted. (They all default to off.) If this is done from the host computer, be sure to set Remote Mode to on in the escape sequence, or the terminal will be in Local Mode, and therefore unable to process any further data from the host computer.

The general format of the terminal configuration  $\text{E}_c$  &q sequence is as follows:

$\text{E}_c$  &q 4menu{—t

[<lock/unlock>1]

[d] (initially sets all menu fields to default values)

[e] (signals start of individual field definitions)

$\text{E}_c$  &q <x>t <y>L (locks or unlocks menu “x” where “x” and “y” are as follows:

<b>“x”</b>	<b>Menu</b>
<b>1</b>	Datacomm
<b>2</b>	Invalid
<b>3</b>	Invalid
<b>4-8</b>	Terminal Configuration
<b>9</b>	Service Keys (valid only in lock parameter)
<b>“y”</b>	Action
<b>0</b>	Unlock
<b>1</b>	Lock

These  $E_c$  sequences are used to change the Configuration Operations, Terminal Configuration menu, Datacomm menu and External Device menu entry values. The active values are changed as well as those in non-volatile memory.

		<x> value	
$E_c \& q < m > te 1 \{ < x > A$	Auto line feed	0 = OFF, 1 = ON	Modes Labels
$E_c \& q < m > te 1 \{ < x > B$	Block mode	0 = OFF, 1 = ON	
$E_c \& q < m > te 1 \{ < x > M$	Modify	0 = OFF, 1 = ON	
$E_c \& q < m > te 1 \{ < x > R$	Remote	0 = OFF, 1 = ON	
$E_c \& q < m > te 1 \{ < x > C$	Caps Lock	0 = OFF, 1 = ON	Terminal Config. Menu
$E_c \& q 8 te 0 \{ < x > D$	Bell	0 = OFF, 1 = ON	
$E_c \& q < m > te 1 \{ < x > L$	Local Echo	0 = OFF, 1 = ON	
Esc & q < m > te 0 { < x > A	XmitFnctn (A)	0 = NO, 1 = YES	
Esc & q < m > te 0 { < x > B	SPOW (B)	0 = NO, 1 = YES	
Esc & q < m > te 0 { < x > C	InhEolWrp (C)	0 = NO, 1 = YES	
Esc & q < m > te 0 { < x > D	Line/Page (D)	0 = LINE, 1 = PAGE	
Esc & q < m > te 0 { < x > G	InhHndShk (G)	0 = NO, 1 = YES	
Esc & q < m > te 0 { < x > H	InhDC2 (H)	0 = NO, 1 = YES	
Esc & q < m > te 0 { < x > N	EscXfer (N)	0 = NO, 1 = YES	
Esc & q < m > te 2 { < x > S	StartCol	1-80	Format Mode
Esc & q < m > te 2 { < x > L	FormsBufSize	0-255	
Esc & q < m > te 2 { < x > F	FldSeparator	0-127	
Esc & q < m > te 2 { < x > R	Blk Terminator	0-127	
Esc & q 8te 1 { < x > R	RETURN = ENTER	0 = NO, 1 = YES	
Esc & q 8te 1 { < x > A	RETURN Def (first character)	See note	
Esc & q 8te 1 { < x > B	RETURN Def (second character)	See note	
Esc & q 8te 1 { < x > T	Tab = Spaces	0 = NO, 1 = YES	
$E_c \& q 8te 1 \{ < x > U$	DATA COMM/PRINTER	0 = PORT1/PORT2 1 = PORT2/PORT1	
Esc & q < m > te 2 { < x > X	Decimal Type	0 = US, 1 = EUR	External Device
Esc & q < m > te 2 { < x > Y	Implied Dec Digits	0-9	
Esc & q < m > te 2 { < x > Z	Transmit	0 = All Fields 1 = Modified fields	
Esc & q 8te 1 { < x > N	Printer Nulls	"x" = no. of nulls (0-255)	

Note: "x" indicates the decimal value of the ASCII code for the desired character.

## Datacomm Menu (HP 2394A)

This method of configuring the datacomm portion of the terminal should be used cautiously when the escape sequence originates from a host computer. If you inadvertently misconfigure the port through which you are connected to the terminal the data link could be disabled.

The general format of the  $\text{E}_c\&q$  sequence is as follows:

```

 $\text{E}_c\&q$  <type>t
  [<lock/unlock>1]
  [<menu type>d]
  [e]
  0{
    [<BaudRate>e]
    [<DataBits>h]
    [<Parity>i]
    [<ChkParity>j]
    [<EnqAck>n]
    [<SR(CH)>p]
    [<Asterisk>q]
    [<CS(CB)Xmit>b]
    [<XmitPace>g]
    [<RecvPace>h]
  }

```

The various parameter values are as follows:

	Point-to-Point
<BaudRate>e	0 = 110 1 = 134.5 2 = 150 3 = 300 4 = 600 5 = 1200 6 = 1800 8 = 2400 9 = 4800 10 = 9600 11 = 19200
<DataBits>h	0 = 7 (valid only if parity is enabled) 1 = 8 (requires parity=none)
<PARITY>i	1 = 0's 1 = ODD 2 = 1's 3 = EVEN 4 = None (valid only with 8-bit data)

<ChkParity>j	0 = YES
	1 = NO
<EnqAck>n	0 = NO
	1 = YES
<SR(CH)>p	0 = LO
	1 = HI
<Asterisk>q	Point-to-Point
	0 = CS
	1 = DM
	2 = OFF
	Internally
	0 = OFF
<CS(CB)Xmit>b	1 = CS
	2 = DM
<CS(CB)Xmit>b	0 = NO
	1 = YES
<XmitPace>g	0 = None
	1 = XonXoff
<RecvPace>h	0 = None
	1 = XonXoff
	2 = TR(CD) (Not valid for full duplex modem)



# Keyboard Control

SECTION

III

## INTRODUCTION

The terminal keyboard is a separate unit that is linked to the display portion of the terminal by a flexible cable. Except for two keys ([Reset Break] and [Stop]), the overall keyboard can be logically divided into a character set group, a numeric pad, a cursor control group, an edit group, and a function key group. The function key group includes eight keys marked [f1] through [f8] and the keys [Menu] and [User System]. The [f1] through [f8] keys are multi-purpose keys in that the functions they perform vary from one situation to another. At any given time the applicable labels from the function keys are displayed across the bottom of the screen (Figures 3-1 and 3-2).

## SELECTING THE MODE

Pressing keys [User System], then [f4] enables the mode selection keys and changes the [f1] through [f8] screen labels to the following:

[f1]	[f2]	[f3]	[f4]	[f5]	[f6]	[f7]	[f8]
LINE MODIFY	MODIFY ALL	BLOCK MODE	REMOTE MODE	SMOOTH SCROLL	MEMORY LOCK	DISPLAY FUNCTNS	AUTO LF

These keys act as toggle switches, in that they alternately enable and disable the designated mode. When a particular mode is enabled, an asterisk is displayed in the label.

### Line Modify Mode ([f1])

When the terminal is in remote mode and character mode, and you are communicating interactively with a host computer, you may sometimes enter an erroneous command string to which the computer responds with an error message. If the command string is a lengthy one and the error consists of only a few characters, it is a nuisance to have to retype the entire string. In such a case, you may instead enable line modify mode (which temporarily switches the terminal to a special form of block mode). You may then move the cursor to the erroneous line on the display and correct the command string. When the string is edited to your satisfaction, you retransmit the line to the host computer by pressing either the [Return] key or the [Enter] key.

Note that while line modify mode results in a block transmission, it is completely independent of the block mode function described on Page 3-3 in this section (you do NOT have to first enable block mode). In fact, line modify mode is a feature that was specifically designed for use when the terminal is operating in character mode.

From the keyboard, you enable line modify mode using the **LINE MODIFY** key. Line modify mode is automatically disabled when you press either **[Return]** or **[Enter]**. If you change your mind and wish to disable line modify mode before retransmitting the command string, press the **LINE MODIFY** key again and the terminal will return to normal character mode.

The terminal remembers which character was the first (leftmost) one that you entered through the keyboard. This means that when you retransmit a line in modify mode, only the keyboard entry portion of the line (the entire edited command string) is retransmitted; any prompt characters preceding the command string are ignored by the terminal. For more detailed information about this feature refer to the discussion of the **Start Col** field of the terminal configuration menu in Section II.

## Modify All Mode ([f2])

When the terminal is in character mode, you can enable modify all mode, which switches the terminal to a special form of block mode. Modify all mode is the same as line modify mode except that it is NOT disabled when you press **[Return]** or **[Enter]**.

From the keyboard, you enable and disable modify all mode using the **MODIFY ALL** key.

From a program executing in a host computer, you enable and disable modify all mode using the following escape sequences:

ENABLE:  $E_C \& k 1M$   
DISABLE:  $E_C \& k 0M$

A modify all mode designator is maintained in non-volatile memory. When you change modes using the **MODIFY ALL** key, you also alter that mode designator in non-volatile memory. When you change modes using the escape sequence, however, the designator is NOT altered.

After a hard reset or turning off the power, the terminal reverts to the mode specified by the modify all designator in non-volatile memory.

### NOTE

Modify mode can be used with the default terminal configuration parameters (see Table 2-1). For instance: when using modify mode you will usually want the data block (not a  $D_1$  handshake control code) to be sent when you press **[Return]** or **[Enter]**, and the default configuration parameters disable the  $D_1/D_2/D_1$  handshake. Also when moving the cursor to the erroneous line you will normally not want the cursor movement escape sequences to be transmitted to the computer (as these will be interpreted as new data, and cause a new error message), and the default configuration parameters disable the transmission of escape sequences.

The modify all and modify line functions are ignored if the terminal is in format mode or block mode.

## Character/Block Mode ([f3])

When the terminal is connected on-line to a remote host computer, it operates in either of the following data transmission modes:

- **Character Mode.** In this mode, data is transmitted a character at a time as it is entered through the keyboard. Control codes (such as  $C_R$  and  $I_F$ ) are also transmitted.
- **Block Mode.** In this mode, data is NOT transmitted at the time it is entered through the keyboard. Instead, you transmit an entire block of data by first typing the data (after initially typing the data, you can move the cursor around and edit the data as desired) and then pressing the [Enter] key.

When the terminal is in block mode, control codes (such as  $C_R$  and  $I_F$ ) are acted upon locally but NOT transmitted with the data block.

From the keyboard, you enable and disable block mode using the **BLOCK MODE** ([f3]) key.

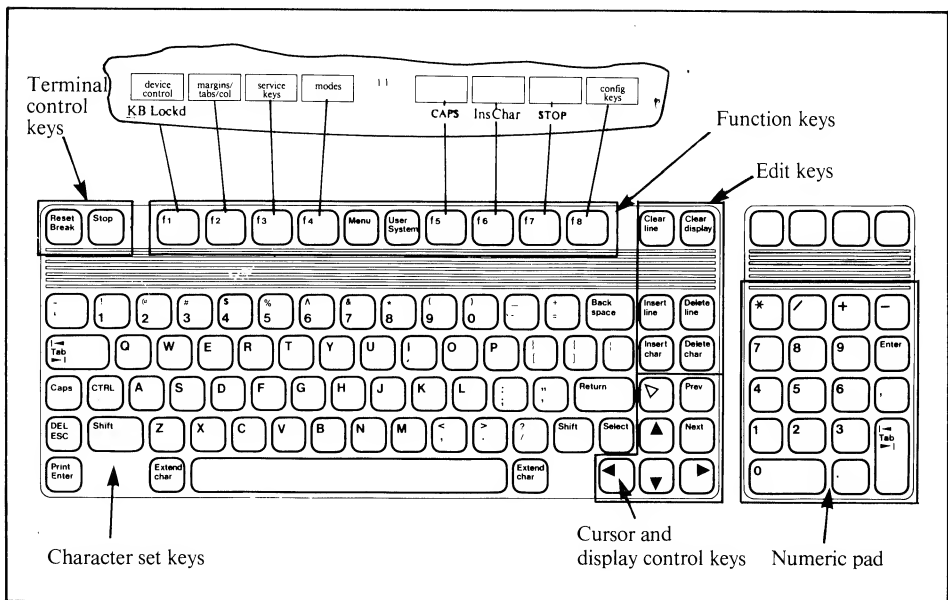


Figure 3-1. Screen-labeled Function Keys of the HP 2392A

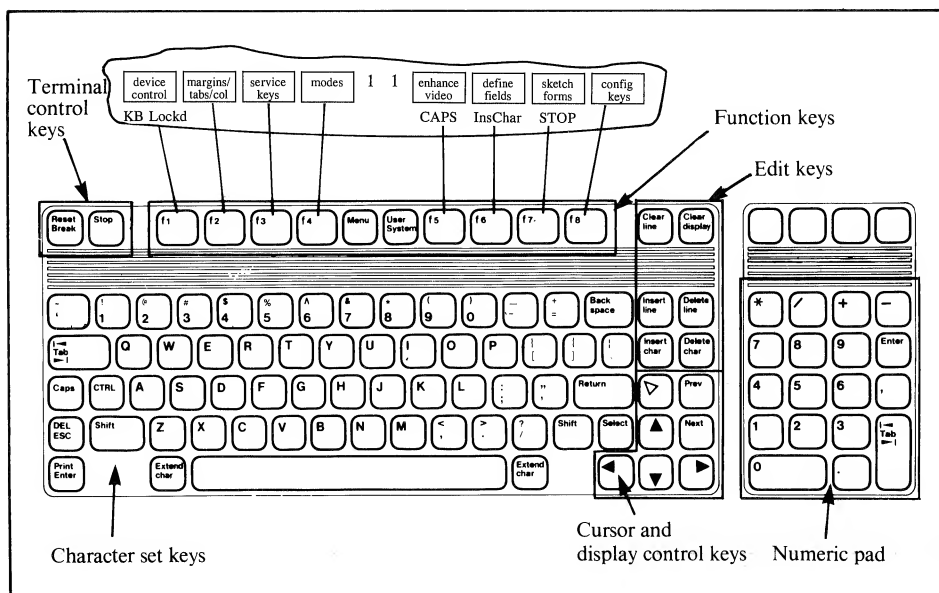


Figure 3-2. Screen-labeled Function Keys of the HP 2394A

From a program executing in a host computer, you enable and disable block mode using the following escape sequences:

ENABLE:  $\text{E}_\text{C}\&\text{k } 1\text{B}$

DISABLE:  $\text{E}_\text{C}\&\text{k } 0\text{B}$

A character/block mode designator is maintained in non-volatile memory. When you change modes using the **BLOCK MODE** key, you also alter that mode designator in non-volatile memory. When you change modes using the escape sequences, however, the designator is NOT altered.

After a hard reset or turning off the power, the terminal reverts to the mode specified by the character/block designator in non-volatile memory.

The relationship between block, line, page, and format modes is described under **[Enter]** key later in this section.

(Note that in ANSI/EM52 mode the terminal is always forced into character mode, and key **[f3]** is disabled).

### Remote/Local Mode (**[f4]**)

When a communications link exists between the terminal and a remote host computer, the terminal is in either of the following two modes:

- **Remote Mode.** In this mode, when you press an alphanumeric key the associated character code is transmitted to the host computer.
- **Local Mode.** In this mode, when you press an alphanumeric key the associated character is displayed at the current cursor position on the screen (nothing is transmitted to the host computer).

From the keyboard, you switch the terminal back and forth between local and remote modes using the **REMOTE MODE** (**[f4]**) key.

From the keyboard or a user-definable key, you can switch the terminal from local to remote (and vice versa) using the following escape sequences:

Local:  $\text{E}_\text{C}\&\text{k } 0\text{R}$

Remote:  $\text{E}_\text{C}\&\text{k } 1\text{R}$

A remote/local mode designator is maintained in non-volatile memory. When you change modes using the **REMOTE MODE** key, you also alter that mode designator in non-volatile memory. When you change modes using the escape sequences, however, the designator in non-volatile memory is NOT altered.

After a hard reset or turning off the power, the terminal reverts to the mode specified by the remote/local designator in the non-volatile memory.

### Smooth Scroll Mode (**[f5]**)

When smooth scroll is enabled, rolling data up and down the screen (using **[Shift] [▲]** and **[Shift] [▼]**) is done smoothly, rather than “jumping” a line at a time.

A smooth scroll mode designator is maintained in non-volatile memory. When you change modes using the smooth scroll key, you can also alter the mode designator in non-volatile memory. When you change modes using the escape sequence, however, the designator is not altered.

From a program executing in a host computer, you enable and disable smooth scroll mode using the following escape sequences:

ENABLE:  $\text{E}\text{C}\&\text{k } 1[$

DISABLE:  $\text{E}\text{C}\&\text{k } 0[$

Note that smooth scroll mode should not be used at baud rates above 2400 without some form of data transfer pacing, such as handshaking. Doing so may result in data loss from overflowing the data buffer.

## Memory Lock Mode ([f6])

Memory lock mode provides two separate functions: overflow protect and display lock.

**Overflow Protect.** This feature prevents you from losing data when display memory is full. If you move the cursor to the first line on the screen and then enable memory lock mode, display memory becomes "protected" so that no data can be lost off the top. In such a case, when you have used all available lines in display memory, any attempt to use more memory is rejected with the error message "MEMORY FULL Press RETURN to clear". You may, however, use the cursor control keys to go back and alter any of the existing data. To continue entering new data, merely disable memory lock mode and reposition the cursor immediately below the last line. Before doing so you may wish to enable data logging (described in Section V) so that data that is then forced off the top of display memory will be retained in printed form.

**Display Lock.** If you position the cursor below the top line of the screen and then enable memory lock mode, the lines above the cursor become "locked" on the screen. As the screen becomes full, the locked lines remain on the screen while subsequent lines roll past the locked rows. This allows you to retain column headings or instructions on the screen as you continue to enter new data. It also provides a useful means of changing the sequence of text blocks as follows:

- a. Press  $[\nabla]$ , [Clear display], and then type the following data:
  3. This is paragraph 3. It should be the third one.
  1. This is paragraph 1. It should be the first one.
  2. This is paragraph 2. It should be the second one.
  4. This is paragraph 4. It should be the last one.
- b. Position the cursor in the first line of paragraph 1.
- c. Enable memory lock mode.
- d. Press keys [Shift]  $[\blacktriangle]$  (roll up) until the first line of paragraph 4 is in the same line as the cursor.

e. Disable memory lock mode and home the cursor. The display should appear as follows:

1. This is paragraph 1. It should be the first one.
2. This is paragraph 2. It should be the second one.
3. This is paragraph 3. It should be the third one.
4. This is paragraph 4. It should be the last one.

From the keyboard, you enable and disable memory lock mode using the **MEMORY LOCK** key. The rows above the line containing the cursor are locked.

Normal editing can be performed within the locked rows; that is, the rows are locked by row number only, so if lines are inserted among the locked rows, they become locked but the total number of locked rows does not increase.

From a program executing in a host computer, you enable and disable memory lock mode using the following escape sequences:

ENABLE:  $E_C I$   
 DISABLE:  $E_C m$

Once enabled, memory lock mode remains enabled until explicitly disabled, until a hard reset is performed, or until the power is turned off.

## Display Functions Mode ([f7])

When display functions mode is enabled the terminal operates as follows:

- In local mode, it displays control codes and escape sequences but does not execute them. For example, if you press the [◀] key the terminal displays  $E_C D$  on the screen but does not perform the “cursor left” function.
- In remote mode, it transmits control codes and escape sequences but does not execute them locally. For example, if you press [Shift] and [▲] (roll up) the terminal transmits an  $E_C S$  but does not perform the “roll up” function. If local echo is enabled (ON) then the  $E_C S$  is also displayed on the screen.

There are two exceptions to the foregoing descriptions:

1. When you press the **DISPLAY FUNCTNS** key, the  $E_C Z$  (which disables display functions mode) or  $E_C Y$  (which enables display functions mode) is executed but is not transmitted or displayed.
2. A  $C_R$  (or  $C_R I_F$  if auto line feed mode is enable) is transmitted, and (if echoed) it is executed and displayed and the terminal also performs a line feed.

From the keyboard, you enable and disable display functions mode using the **DISPLAY FUNCTNS** key.

From a program executing in a host computer, you enable and disable display functions mode using the following escape sequences:

ENABLE:  $E_C Y$   
 DISABLE:  $E_C Z$

#### NOTE

There is interaction between display functions and the **Xmit=Functn (A)** field of the terminal configuration menu. If **XmitFunctn (A)** is on (YES), the **DISPLAY FUNCTNS** key transmit  $E_C Y$  but not  $E_C Z$ .

Once enabled, display functions mode remains enabled until explicitly disabled, until a soft or hard reset is performed, or until the power is turned off.

### Auto Line Feed Mode ([f8])

When auto line feed mode is enabled, a line feed control code is automatically appended to each carriage return control code generated through the keyboard. That is, every  $C_R$  code generated through the keyboard becomes a  $C_R L_F$ .

Carriage return control codes can be generated through the keyboard in any of the following ways:

- By pressing the [Return] key, provided that a  $C_R$  code is included in the key definition.
- By simultaneously pressing the [CTRL] and [M] keys.
- By pressing any of the user keys ([f1] through [f8]), provided that a  $C_R$  code is included in the particular key definition.
- By pressing the [Enter] key when the terminal is in block mode, line modify mode, or modify all mode (in these cases a  $C_R$  code is transmitted as the line terminator, see table 3-1)

From the keyboard, you enable and disable auto line feed mode using the **AUTO LF** key.

From a program executing in a host computer, you enable and disable auto line feed mode using the following escape sequences:

ENABLE:  $E_C \&k 1A$   
 DISABLE:  $E_C \&k 0A$

When you enable or disable auto line feed mode using the “**AUTO LF**” key, you also alter the content of the “**AUTO LF**” field in both active and non-volatile memory. When you enable or disable auto line feed mode using the escape sequence, however, you only change the content of the “**AUTO LF**” field in active memory.

After a hard reset or turning off the power, the terminal reverts to the mode specified by the “**AUTO LF**” field in non-volatile memory.



## Caps Mode ([Caps] key control)

When caps mode is enabled, all unshifted alphabetic keys generate uppercase letters and all shifted alphabetic keys generate lowercase letters. This mode is used primarily as a typing convenience and only affects the 26 alphabetic keys.

From the keyboard, you enable and disable caps mode using the [Caps] key. This key alternately enables and disables caps mode. On USASCII keyboard, the [Caps] key has no effect if “Caps Lock” in terminal configuration is enabled (ON).

From a program executing in a host computer, you enable and disable caps mode using the following escape sequences:

ENABLE:  $E_C \& k 1P$

DISABLE:  $E_C \& k 0P$

Once enabled, caps mode remains enabled until explicitly disabled, until a hard reset is performed, or until the power is turned off.

When enabled, “CAPS” is displayed in the status line.

## Caps Lock Mode (Caps Lock menu control)

The caps lock mode only affects a USASCII keyboard (the “Caps Lock” field is ignored with a national keyboard).

When caps lock mode is enabled, the terminal generates only Teletype-compatible codes: uppercase ASCII (00-5F, hex) and DEL (7F, hex). Unshifted alphabetic keys (a-z) generate the codes for their uppercase equivalents, the {, |, and } keys generate the codes for [, \, and | (respectively), and the ` and ~ keys are ignored.

From the keyboard, you enable and disable caps lock mode using the “Caps Lock” field of the terminal configuration menu described in Section II.

From a program executing in a host computer, you enable and disable caps lock mode using the following escape sequences:

ENABLE:  $E_C \& k 1C$

DISABLE:  $E_C \& k 0C$

At any given time the current state (enabled/disabled) of caps lock mode is reflected in the “Caps Lock” field of the terminal configuration menu. When you enable or disable the mode by altering the menu field from the keyboard and then pressing the **SAVE CONFIG** key, you alter both the active and non-volatile memory versions of that field. When you enable or disable the mode using the escape sequence, however, you only change the active value of the “Caps Lock” field in the terminal configuration menu.

After a hard reset or turning off the power, the terminal reverts to the mode specified by the “Caps Lock” field in the terminal configuration menu in non-volatile memory.

## Send Cursor Position Mode

Some programs need to know where the cursor is located before it is moved to the beginning of the data to be transmitted. Other application programs may need to do some other operation before sending data. Currently some application programs get the cursor location by sending a cursor sense escape sequence ( $E_Ca$ ) after the terminal has sent a DC2. The terminal will then send the cursor address, the host sends a DC1, and the terminal then sends the data. But this does not work when handshaking is disabled.

Therefore, the Send Cursor Position (SCP) mode has been created. It is accessible by escape sequence only.

ENABLE:  $E_C \&x 1C$

DISABLE:  $E_C \&x 0C$

When disabled, the terminal will act as previously defined. When enabled, and the [Enter] key or [Select] key or a user defined key set to transmit (T) is pressed, the cursor position in the form of  $E_C \&axxcyyyR$  is inserted at the beginning of the block sent to the computer. The position is of the same form as that returned from an absolute cursor position sense status request ( $E_Ca$ ) described in Section IV. The position value is that of the cursor BEFORE any repositioning is done, regardless of strap settings. The block transfer obeys the usual handshaking conventions.

The Send Cursor Position Mode does NOT apply to the [Enter] and [Return] keystrokes for line modify and modify all modes. These modes are defined only when in character mode and are intended to simulate a block mode operation.

A soft reset will leave the Send Cursor Position Mode unchanged. A hard reset will set the Send Cursor Position Mode to its default (disabled) setting.

## Format Mode

The terminal includes a format mode in which elaborate, custom-designed forms containing protected and unprotected fields can be displayed on the screen and used for data entry.

When format mode is enabled, the terminal operator may only enter data into unprotected fields. If the operator positions the cursor in a protected area and then attempts to type data, the cursor automatically moves to the start of the next unprotected field before the terminal accepts the data.

The designing of forms and the use of format mode are described in Section IV.

From a program executing in a host computer or from the keyboard, you enable and disable format mode using the following escape sequences:

ENABLE:  $E_C W$

DISABLE:  $E_C X$

Once format mode is enabled, it remains enabled until explicitly disabled, until a hard reset is performed, or until the power is turned off.

## Auto-keyboard Lock Mode

When a terminal is connected to a packet switching network (using X.25 protocol) via a controller/multiplexer, it is necessary to ensure that the packet sent is received and acted upon before another is sent (from the terminal). In order to achieve this, the keyboard must automatically lock such that it can only be unlocked by the receiving host. This is the Auto-keyboard Lock. This mode is accessible by escape sequence only.

When disabled (default), the terminal acts normally.

When enabled, any user key configured to be transmit only (T) or the [Enter] key or [Select] key causes the keyboard to be locked after the data transfer has taken place and the message "KB Lockd" to appear in the screen Status Line. The host computer can then unlock the keyboard (with  $E_Cb$ ) when it is ready to receive more data; when the keyboard is unlocked, the keyboard buffer is flushed (cleared).

A soft Reset will not affect the auto-keyboard lock mode, but will still unlock the keyboard. A hard Reset or Power-on will put this mode in its default state (disabled).

Note that the Send-display sequence ( $E_Cd$ ) does not simulate the [Enter] keystroke, and is not affected by the auto-keyboard lock mode (i.e. it does not lock the keyboard). The function key triggering sequence ( $E_C \& f < \text{key} > E$ , where  $< \text{key} >$  is the function key number) is defined to act as though the key was depressed, so it is affected by this mode (i.e. it locks the keyboard).

From a program executing in a host computer, you enable and disable auto-keyboard lock mode using the following escape sequences:

ENABLE:  $E_C \& k 1K$   
 DISABLE:  $E_C \& k 0K$

## USER-DEFINABLE KEYS

The eight function keys ([f1] through [f8]), besides performing their usual terminal control functions, can be defined either locally by the terminal operator or remotely by a program executing in a host computer. By "defined" it means that:

1. You can assign to each key a string of ROMAN 8 alphanumeric characters and/or control codes (such as  $C_R$  or  $I_F$ ).
2. You can specify each key's operational attribute: whether its content is to be executed locally at the terminal, transmitted to a host computer, or both.
3. You can assign to each key an alphanumeric label (up to 16 characters) which, in user keys mode, is displayed across the bottom of the screen.

When defining a key from the keyboard, the key content may include explicit escape sequences (entered using display functions  $mod_c$ ) that control or modify the terminal's operation (when the key is subsequently used).

The definition of each user key may contain up to 80 characters (alphanumeric characters, control characters, and explicit escape sequence characters). The definition whether entered locally through the keyboard or received from a remote computer is saved after power off in the non-volatile memory, but reset to  $E_Cp-E_Cw$  following a hard reset from the host computer using  $E_CE$ .

### NOTE:

The HP 2394A terminal offers 8 user-definable function key definitions **plus** the 8 permanent definitions (the default values) see "Defining Keys Locally."

## Defining Keys Locally

To define one or more keys from the keyboard, press the [CTRL] and [Menu] keys simultaneously (or  $E_CJ$ ). The user keys menu shown in figure 3-3 then appears on the screen. Note that then menu in Figure 3-3 contains the default values for all of the fields. While the menu is displayed on the screen, you can reset the entire menu to the default values by pressing the **DEFAULT VALUES** function key ([f4]).

### NOTE

On the HP 2394A terminal, these function key default values can be invoked by pressing [Shift] [f1] to [Shift] [f8].

The menu contains a set of unprotected fields that you access using the [Tab▶◀] key.

For each user key the menu contains four unprotected fields:

- **Attribute Field.** This one-character field always contains an uppercase L, T or N signifying whether the content of the particular user key is to be:
  - a. Executed locally only (L).
  - b. Transmitted to the host computer only (T).
  - c. Treated as though it were entered through the keyboard (N).

The alphanumeric keys are disabled when the cursor is positioned in this field. You change the content of this field by pressing the **NEXT CHOICE** and **PREVIOUS CHOICE** keys ([f2] and [f3], respectively).

- **Two Label Fields.** The two eight-character fields to the right of the word "LABEL" allow you to supply the user key's label. When the terminal is in user keys mode, the key labels are displayed from left to right in ascending order across the bottom of the screen (each displayed key label occupies two lines). The first **LABEL** field in the user keys menu supplies the upper portion of the particular key label while the second supplies the lower portion. The HP 2394A provides the ability to include video enhancements, e.g. half-bright, blinking, etc., within each label. (See Section IV: Enhance Video Set.)
- **Key Definition Field.** The entire line (80 characters) immediately below the attribute and label fields is available for specifying the character string that is to be displayed, executed, and/or transmitted whenever the particular key is physically pressed. When entering characters into this field you may use display functions mode to display control characters. Note that when the menu is displayed the alternate line drawing set and display enhancements are inhibited. (See "Two Label Fields" above.)

When entering the label and key definition you may access display functions mode by way of the **DISPLAY FUNCTNS** function key ([f7]). Note that this implementation of display functions mode is separate from that which is enabled/disabled via the mode selection keys.

The [Return] key can be used for including  $C_R$  codes (with display functions mode enabled) in key definitions. If auto line feed mode is also enabled, the [Return] key will generate a  $C_R L_F$ , otherwise it is considered a cursor movement key.

When the user keys menu is displayed on the screen you may use the [Insert char], [Delete char], and [Clear line] keys for editing the content of the label and key definition fields.

When you have finished defining all the desired keys, press the **[User System]** key (the user keys menu disappears from the screen). When you press **[Shift] [User System]** or **[CTRL] [User System]**, or **[Menu]** twice, or enter  $E_{CJB}$ , the defined user key labels are displayed across the bottom of the screen and the **[f1]** through **[f8]** user keys, as defined by you, are enabled.

## Defining Keys Programmatically

From a program executing in a host computer, you can define one or more keys using the following escape sequence format:

$$E_{\text{C\&F}} \langle \text{attribute} \rangle \text{ a } \langle \text{key} \rangle \text{ k } \langle \text{label length} \rangle \text{ d } \langle \text{string length} \rangle \text{ L } \langle \text{label} \rangle \langle \text{string} \rangle$$

where:

<attribute> = 0: normal (0 is the default)

1: local only

2: transmit only

**<key>** = 1-8: f1 - f8, respectively (1 is the default)

<label length> = 0-16 (0-32 diacritical combinations) (0 is the default)

<string length> = 0-80 (0-160 diacritical combinations) (1 is the default)

(-1 causes field to be erased)

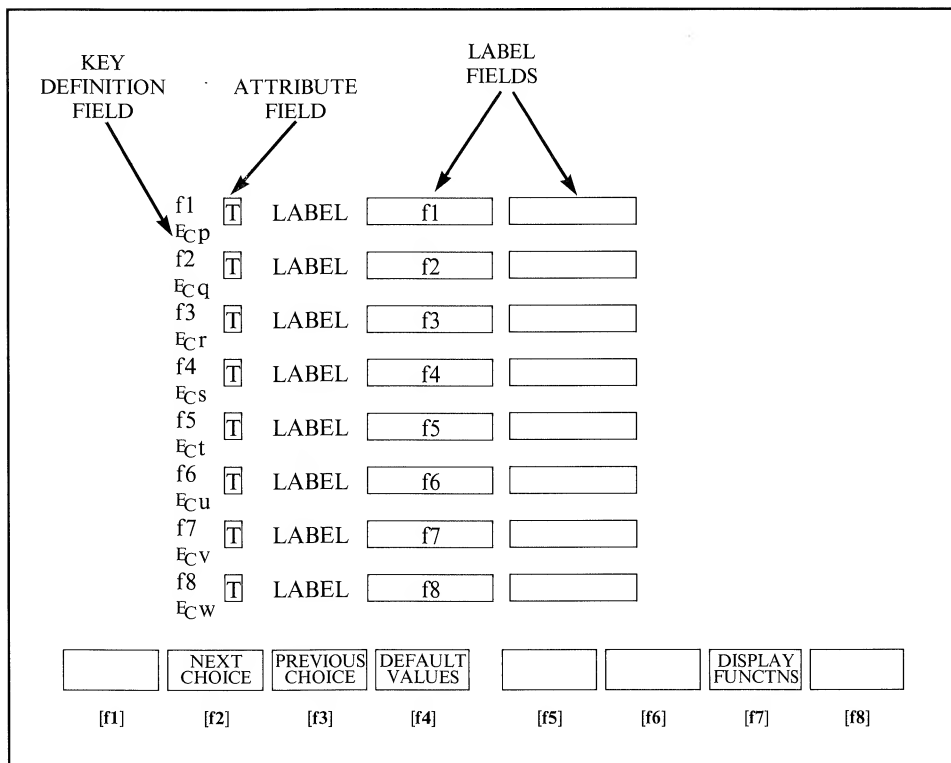


Figure 3.3 User Keys Definition Menu and Default User Key Labels

- <label> = the character sequence for the label field
- <string> = the character sequence for the key definition field

The <attribute>, <key>, <label length>, and <string length> parameters may appear in any sequence but must precede the label and key definition strings. You must use an uppercase identifier (A, K, D, or L) for the final parameter and a lowercase identifier (a, k, d, or l) for all preceding parameters. Following the parameters, the first 0 through 16 characters, as designated by <label length>, constitute the key's label and the next 0 through 80 characters, as designated by <string length>, constitute the key's definition string. The total number of displayable characters (alphanumeric data, control codes such as  $\text{C}_R$  and  $\text{I}_F$ , and explicit escape sequence characters) in the label string must not exceed 16, and in the definition string must not exceed 80.

#### NOTE

This (and the lock configuration menus sequence) are the only escape sequences to be retained on non-volatile memory.

One function key may be used to trigger another using the " $\text{E}_C\&\text{f}<\text{key}>\text{E}$ " sequence. However, only one key may be triggered. That is, only one " $\text{E}_C\&\text{f}<\text{key}>\text{E}$ " sequence may be included in a key's <string>.

Example: Assign **LOG-ON** (6 characters) as the label and **HELLO USER.ACCOUNT**  $\text{C}_R$  (19 characters) as the definition for the [f5] user key, and enable the user keys. The key is to have the default attribute "N".

$\text{E}_C\&\text{f5k6d19LLOG-ONHELLO USER.ACCOUNT}_R\text{E}_C\&\text{jB}$

After issuing this escape sequence from your program to the terminal, the [f5] portion of the user keys menu is as follows:

```
f5 [N] LABEL LOG-ON [ ]
HELLO USER.ACCOUNT  $\text{C}_R$ 
```

If the transmit only attribute (2) is designated, the particular user key will have no effect unless the terminal is in remote mode. A transmit only user key may (when subsequently pressed) invoke a block transfer handshake and append the appropriate terminator to the string. (When send cursor position mode is enabled, the cursor position is inserted at the beginning of the block sent to the computer, see Page 3-10). The  $\text{E}_C\&\text{jB}$  sequence turns on the user labels.

Note that in remote mode when the user key definition menu is displayed, pressing [Enter] or sending escape sequence  $\text{E}_C\text{d}$  (transmit a block) can be used to transfer a copy of the menu to the computer. The returned escape sequence is identical to the one used by the host to configure the user keys ( $\text{E}_C\&\text{f}...$ ). In addition, in block page mode, eight escape sequences are returned (one for each of the eight keys). In character mode and block line mode, only the escape sequence for the user key in whose field the cursor is located is returned. For example, if the default menu is displayed with the cursor in one of the fields of key [f1], the following escape sequence is returned to the computer in character mode or block line mode:

$\text{E}_C\&\text{f2a1k16d2L}$  f1  $\text{E}_C\text{p}$

## Controlling The User Keys Menu Programmatically

From a program executing in a host computer, you can display the user keys menu on the screen and remove it from the screen using the following escape sequences:

DISPLAY MENU:  $\text{E}_C\text{j}$   
 REMOVE MENU:  $\text{E}_C\text{k}$

## Controlling The Function Key Labels Programmatically

From a program executing in a host computer, you can control the function key labels display as follows by using escape sequences:

- You can remove the key labels from the screen entirely (this is the equivalent of pressing the [Menu] key).
- You can enable the mode selection keys (this is the equivalent of pressing the [User System] key, then [f4] key).
- You can enable the user keys (this is the equivalent of pressing [Shift] and [User System], or [CTRL] and [User System], or [Menu] twice).
- You can “lock” the current set of labels on the screen (i.e. disable the [User System] key and [Menu] key).
- You can reenable the [User System] key and [Menu] key.

The escape sequences are as follows:

$\text{E}_C\&\text{j}@$	Enable the user keys, and remove all key labels and status line from the screen.
$\text{E}_C\&\text{jA}$	Enable and display the “modes” labels.
$\text{E}_C\&\text{jB}$	Enable and display the user function key labels.
$\text{E}_C\&\text{jR}$	Enable keys [User System] and [Menu] and label “modes”.
$\text{E}_C\&\text{jS}$	Disable keys [User System] and [Menu] and label “modes”.
$\text{E}_C\&\text{j} <\text{xx}> \text{L} <\text{message}>$	Remove the key labels from the screen and display the character string $<\text{message}>$ (which consists of $<\text{xx}>$ characters; where $<\text{xx}>$ may be up to 160 characters, 320 diacritical combinations). The terminal beeps at the end of the displayed $<\text{message}>$ . The labels are restored by “ $\text{E}_C\&\text{jC}$ ” or pressing [Return]. Note that all control characters (except $\text{C}_R$ and $\text{I}_F$ ) included in the $<\text{message}>$ are displayed. Also if the number of characters in the $<\text{message}>$ is less than the specified $<\text{message length}>$ , characters entered on the keyboard (if echoed) will be displayed on the labels line until the specified number is reached.
$\text{E}_C\&\text{jC}$	Remove your $<\text{message}>$ from the screen and restore the current key labels.

## ENABLE/DISABLE KEYBOARD

You can enable and disable the terminal's keyboard by executing escape sequences. When the keyboard is disabled all keys EXCEPT the following are ignored:

[Shift], [CTRL], [Reset Break]

The escape sequences for enabling and disabling the keyboard are as follows:

ENABLE:  $E_{Cb}$

DISABLE:  $E_{Cc}$

Once disabled, the keyboard remains disabled until explicitly enabled, until a soft or hard reset is performed, or until the power is turned off.

When the keyboard is disabled, the "KB Lockd" indication appears in the status line.

### [Print Enter] KEY

#### Enter

When the terminal is in remote mode, pressing the [Print Enter] key on its own sends a pending block transfer of data from display memory to host computer. This locks the keyboard (causing the message "KB Lockd" to appear in the display's status line) until the resultant data transfer is complete or until the host unlocks the keyboard if auto-keyboard lock mode is on.

When the Send Cursor Position Mode is enabled, and the [Enter] key is pressed, the cursor position is inserted at the beginning of the block sent to the host computer (see Send Cursor Position Mode).

The type of handshaking used and precisely what data is transmitted depends on the following factors:

1. Whether the terminal is in character mode, block line mode, or block page mode.
2. Whether or not the terminal is in format mode.
3. The settings of the **InhHndShk (G)**, **Inh DC2 (H)**, and **line/page** fields in the terminal configuration menu.

Table 3-1 summarizes the effect of the [Enter] key in each of the possible mode/strap combinations. (Sections VI details the handshake operations).

#### Print

Pressing the [Print Enter] key with [Shift] causes the cursor to be "homed up" and the display memory to be copied to the external printer (if installed). If no "to" device has been specified the message "No "TO" device Press RETURN to clear" is displayed. If only the display was set as the "to" device the message "Source = Destination Press RETURN to clear" is displayed. If a printer module is not installed, no action is taken.



## SEND DISPLAY ( $E_{Cd}$ )

From a program executing in a host computer, you can trigger a block transfer of data from display memory to the host computer by issuing the following escape sequence:

$E_{Cd}$

This escape sequence is only responded to when received over a datacomm line; it is ignored if entered through the keyboard or issued from a user key (unless block mode is enabled). With the following two exceptions, the resultant data transfer is performed as though the **[Enter]** key had been pressed:

1. The cursor is NOT repositioned. The data transfer always begins at the current cursor position.
2. The type of handshaking used is determined as follows (see Section VI for details):

InhHndShk (G) = NO	} $D_1$ handshake
Inh DC2 (H) (ignored)	
InhHndShk (G) = YES	} $D_1/D_2/D_1$
Inh DC2 (H) = NO	
InhHndShk (G) = YES	} $\rightarrow$ no handshake
Inh DC2 (H) = YES	

The  $E_{Cd}$  sequence also temporarily disables the keyboard (with “KB Lockd” in the status line) so that the **[Enter]** key cannot be used until the current data transfer is completed. If the  $E_{Cd}$  sequence is received while an **[Enter]** key data transfer is in progress, the escape sequence is ignored.

Note that an  $E_{Cd}$  sequence resets the “block trigger received” flag. This means, for example, that if you are using the  $D_1$  handshake and the terminal receives a  $D_1$  followed by the  $E_{Cd}$ , it “forgets” that a block trigger was just received and thus will NOT send the data immediately. The terminal must receive another  $D_1$  before it will start the data transfer.

The amount of data transferred depends on the following terminal settings: **page/line** field in terminal configuration, and whether block mode, modify all, or line modify is enabled. For more detailed information, refer to Table 3-1 (**[Enter]** Key Operation).

Table 3-1. [Enter] Key Operation

**CHARACTER MODE**

The cursor is repositioned to left margin.

All characters through the first subsequent block terminator or through the end of the line (whichever is encountered first) are transmitted to the host computer as a block.

Control codes, video enhancement escape sequences, alternate character set escape sequences, and field definition escape sequences are transmitted if encountered.

If the operation is terminated by encountering the end of the line, the terminal sends a  $C_R$  (or a  $C_R I_F$  if auto line feed mode is enabled). The cursor is repositioned to column 1 and a line feed is performed if auto line feed mode is enabled.

If the operation is terminated by encountering a block terminator, the terminal sends a block terminator followed by a  $C_R$  (or a  $C_R I_F$  if auto line feed mode is enabled). The cursor remains positioned immediately following the terminator.

If there is no data to be transmitted, the terminal sends the configured block terminator followed by a  $C_R$  (or a  $C_R I_F$  if auto line feed mode is enabled).

The type of handshaking used is determined as follows:

InhHndShk (G) = YES	} $D_1/D_2/D_1$
Inh DC2 (H) = NO	

Any other combination → no handshake

**Character Mode, Format Mode**

If the cursor is within an unprotected field, all characters from the current cursor position through the end of the field are transmitted to the host computer as a block. Otherwise, the terminal searches for the next subsequent unprotected field and transmits the content of that field.

Control codes within the field are transmitted.

Video enhancement escape sequences, alternate character set escape sequences, and field definition escape sequences within the field are NOT transmitted.

If the operation is terminated by encountering the end of the unprotected field, the terminal sends a  $C_R$  (or a  $C_R I_F$  if auto line feed mode is enabled). The cursor remains at the first character position after the end of the field.

If the operation is terminated by encountering a block terminator, the terminal sends a block terminator followed by a  $C_R$  (or a  $C_R I_F$  if auto line feed mode is enabled). The cursor remains positioned immediately following the terminator.

Continued . . .

Table 3-1. [Enter] Key Operation (Continued)

If there is no data to be transmitted, the terminal sends a block terminator followed by a  $C_R$  (or a  $C_R I_F$  if auto line feed mode is enabled). The  $C_R$  that is transmitted has no effect on the terminal locally, and the cursor remains unmoved.

The type of handshaking used is determined as follows:

InhHndShk (G) = YES	} $D_1/D_2/D_1$
Inh DC2 (H) = NO	
Any other combination → no handshake	

## BLOCK LINE MODE

Block line mode means that block mode is on and that the **line/page (D)** field in terminal configuration is set to "line".

Inh DC2 (H) = YES

The cursor is repositioned to column 1 within the current line. All characters through the first subsequent block terminator or through the end of the line (whichever is encountered first) are then transmitted to the host computer as a block.

Inh DC2 (H) = NO

The cursor is NOT repositioned. All characters through the first subsequent block terminator or through the end of the line (whichever is encountered first) are transmitted to the host computer as a block.

Control codes, video enhancement escape sequences, alternate character set escape sequences, and field definition escape sequences are all transmitted if encountered.

If the operation is terminated by encountering the end of the line, the terminal sends a  $C_R$  (or a  $C_R I_F$  if auto line feed mode is enabled). The cursor is repositioned to column 1 and a line feed is performed if auto line feed mode is enabled.

If the operation is terminated by encountering a block terminator, the terminal sends a block terminator followed by a  $C_R$  (or a  $C_R I_F$  if auto line feed mode is enabled).

The cursor remains positioned immediately following the terminator.

If there is no data to transmit, a block terminator followed by a  $C_R$  or  $C_R I_F$  is transmitted. The cursor is not moved.

The type of handshaking used is determined as follows:

InhHndShk (G) is ignored
Inh DC2 (H) = NO → $D_1/D_2/D_1$
Inh DC2 (H) = YES → no handshake

Continued . . .

Table 3-1. [Enter] Key Operation (Continued)

**Block Line Mode, Format Mode**

Block line mode means that block mode is on and the **line/page (D)** field in terminal configuration is set to "line".

If the cursor is within an unprotected field, all characters from the current cursor position through the end of the field are transmitted to the host computer as a block. Otherwise, the terminal searches for the next subsequent unprotected field and transmits the content of that field.

Control codes within the field are transmitted.

Video enhancement escape sequences, alternate character set escape sequences, and field definition escape sequences within the field are NOT transmitted.

If the operation is terminated by encountering the end of the unprotected field, the terminal sends a  $C_R$  (or a  $C_R I_F$  if auto line feed mode is enabled). The cursor remains positioned at the end of the field.

If the operation is terminated by encountering a block terminator, the terminal sends a block terminator followed by a  $C_R$  (or a  $C_R I_F$  if auto line feed mode is enabled). The cursor remains positioned immediately following the terminator.

If there is no data to be transmitted, the terminal sends a block terminator followed by a  $C_R$  (or a  $C_R I_F$  if auto line feed mode is enabled). The  $C_R$  that is transmitted has no effect on the terminal locally, and the cursor remains unmoved.

The type of handshaking used is determined as follows:

InhHndShk (G) (ignored)

Inh DC2 (H) = NO  $\rightarrow D_1/D_2/D_1$

Inh DC2 (H) = YES  $\rightarrow$  no handshake

**Block Page Mode**

Block page mode means that block mode is on and the **line/page (D)** field in terminal configuration is set to "page".

Inh DC2 (H) = YES

The cursor is repositioned to the "home up" position. All characters through the first subsequent block terminator or through the end of display memory (whichever is encountered first) are transmitted to the host computer as a series of blocks, each block corresponding to one line in display memory.

Inh DC2 (H) = NO

The cursor is NOT repositioned. All characters from the cursor position through the first subsequent block terminator or through the end of display memory (whichever is encountered first) are transmitted to the host computer as a series of blocks. Each block corresponds to one line in display memory.

Continued...

Table 3-1. [Enter] Key Operation (Continued)

Control codes, video enhancement escape sequences, alternate character set escape sequences, and field definition escape sequences are all transmitted if encountered.

After each line (except the final one) the terminal sends a  $C_R I_F$ . If the operation is terminated by encountering the end of display memory, the terminal sends a  $C_R I_F$  followed by a block terminator after the last line. If the operation is terminated by encountering a block terminator, the terminal sends only a block terminator after the last line.

If there is no data to be transmitted, the terminal sends a  $C_R I_F$  followed by the configured block terminator.

The type of handshaking used is determined as follows:

InhHndShk (G) (ignored)

Inh DC2 (H) = NO  $\rightarrow D_1/D_2/D_1$

Inh DC2 (H) = YES  $\rightarrow$  no handshake

### Block Page Mode, Format Mode

Inh DC2 (H) = YES

The cursor is repositioned to the "home up" position. All unprotected characters through the first subsequent block terminator or through the end of display memory (whichever is encountered first) are transmitted to the host computer as a series of blocks. Each block corresponds to one unprotected field.

Inh DC2 (H) = NO

The cursor is NOT repositioned. All unprotected characters through the first subsequent block terminator or through the end of display memory (whichever is encountered first) are transmitted to the host computer as a series of blocks. Each block corresponds to one unprotected field.

Control codes within the fields are transmitted.

Video enhancement escape sequences, alternate character set escape sequences, and field definition escape sequences within the fields are NOT transmitted.

After each field (except the final one), the terminal sends a field separator. After the final field, the terminal sends a block terminator.

If the end of display memory is encountered before locating an unprotected field, the terminal merely sends a block terminator.

The type of handshaking used is determined as follows:

InhHndShk (G) (ignored)

Inh DC2 (H) = NO  $\rightarrow D_1/D_2/D_1$

Inh DC2 (H) = YES  $\rightarrow$  no handshake

Continued . . .

Table 3-1. [Enter] Key Operation (Continued)

**MODIFY MODE**

Note that modify line and modify all modes are functional only when the terminal is configured for character mode operation. When either block mode or format mode is enabled, the [Enter] key operates as described for block mode earlier in this table.

In modify mode, the cursor is repositioned as follows:

1. To the logical start-of-text pointer; or
2. To the designated start column (Start Col) if there is no logical start-of-text pointer.

For more information on the logical start-of-text pointer and start column, refer to Table 2-1, "Terminal Configuration Menu Fields" in Section II.

All characters through the first subsequent block terminator or through the end of the line (whichever is encountered first) are transmitted to the host computer as a block.

Control codes, video enhancement escape sequences, alternate character set escape sequences, and field definition escape sequences are all transmitted if encountered.

If the operation is terminated by encountering the end of the line, the terminal sends a  $C_R$  (or a  $C_R$   $L_F$  if auto line feed mode is enabled). If **LocalEcho** = **OFF**, the cursor is repositioned to the column at which the transmission began, otherwise the cursor is repositioned to column 1. A line feed is performed if auto line feed mode is enabled.

If the operation is terminated by encountering a block terminator, the terminal sends a block terminator followed by a  $C_R$  (or a  $C_R$   $L_F$  if auto line feed mode is enabled).

The cursor remains positioned immediately following the terminator.

The type of handshaking used is determined as follows:

InhHndShk (G) = YES	}	$D_1/D_2/D_1$
Inh DC2 (H) = NO		
Any other combination → no handshake		

The [Enter] key can be triggered from the host by the following sequence:  $E_C$ &f-1E

**[Reset Break] KEY****Soft Reset**

A soft reset does the following:

1. Rings the terminal's bell.
2. Halts any device operations currently in progress.
3. Enables the keyboard (if disabled).
4. Clears any existing error conditions and removes the error message display (if present) from the bottom of the screen.
5. Disables display functions mode (if enabled).
6. Halts any datacomm transfers currently in progress, clears the datacomm buffer. If the datacomm configuration menu "RecvPace" field is set to "XON/XOFF" and an XOFF had been sent (before the soft reset), an XON is sent.
7. Turns off record mode, if on.

The data on the screen, all terminal operating modes (except display functions mode), and all active configuration parameters are unchanged.

From the keyboard, you perform a soft reset by simultaneously pressing the **[Shift]** and **[Reset Break]** keys.

From a program executing in a host computer, you perform a soft reset using the following escape sequence:

E<sub>C</sub>g

**Hard Reset**

A hard reset has the same effect as turning the terminal's power off and then back on; except that the power-on self-test is not performed. A hard reset from the computer using E<sub>C</sub>E will reset the user definable softkeys to their default values (E<sub>C</sub>p, E<sub>C</sub>q to E<sub>C</sub>w).

A hard reset does the following:

1. Rings the terminal's bell.
2. Halts any device operations currently in progress.
3. Enables the keyboard (if disabled).
4. Clears all the display memory.
5. Clears any existing error conditions and removes the error message display (if present) from the bottom of the screen.
6. Halts any datacomm transfers currently in progress, clears the datacomm buffer, and reinitializes the datacomm port according to the appropriate power-on datacomm configuration parameters. If the datacomm configuration menu "RecvPace" field is set to "XON/XOFF", an XON is sent.
7. Resets the terminal configuration menu parameters to values saved in non-volatile memory or to their default power-on values.

8. Resets certain operating modes and parameters as follows:

Disables display functions mode, caps mode, data logging, smooth scroll, memory lock mode, and modify line.

Disables display functions mode, caps mode, data logging, and modify line.

Resets the left margin to column 1.

Resets the right margin to column 80.

Clears all tab stops set via margins/tabs/col screen label (except implicit stop in left margin).

Turns off the "insert character" function edit.

Resets the User Keys to default values (only using hard reset from computer; keys retain pre-set value after hard reset from the keyboard using [Shift] [CTRL] [Reset Break]).

Resets record mode.

From the keyboard, you perform a hard reset by simultaneously pressing the [Shift], [CTRL] and [Reset Break] keys.

From a program executing in a host computer, you perform a hard reset using the following escape sequence:

$E_C E$

### Break

In remote mode, pressing the [Reset Break] key on its own transmits a 200 ms space on the asynchronous "transmit data" communications line. This serves as a "break" signal to interrupt computer operations.

### [Extend char] KEY

Pressing this key allows access to some of the ROMAN 8 characters (National and displaced USASCII) which are not directly accessed on the keyboard. This key is only active if 8 bit transmission is used (any attempt to use this key in 7 bit transmission will cause the beeper to sound), and is a non-latching key.

### [Select] KEY

This key is treated the same as a "Transmit" softkey and sends  $E_C P$  when pressed. It can be used to easily select a field in a form or a menu. Typically, the user will move the cursor to the requisite field, and then press [Select]. See Section VI for handshaking considerations. When send cursor position mode is enabled, the cursor position is inserted at the beginning of the block sent to the computer, see Page 3-10. (Note in ANSI/EM52 mode, the [Select] key is disabled).

The [Select] key can be disabled and enabled by a program operating on a host computer with the following escape sequence:

$E_C k <x>]$

where:  $<x> = 0$ : disabled  
 $= 1$ : enabled

The [Select] key is disabled (default value) upon Power on and after Hard Reset.

### [Return] Key

The HP 2394A will allow a program running in a host computer to trigger the execution of the RETURN key. The escape sequence used for this purpose is:

$E_C f0 E$



This type of escape sequence is also used to trigger execution of the user function keys, i.e.,  $E_C\&f1E$  to  $E_C\&f8E$ . This sequence is allowed once within a function key definition at the end of the definition to effectively concatenate two or more key definitions into one.

## [Stop] KEY

Alternatively stops and resumes the transfer of data from the datacomm input buffer to the screen. This key is executed only if the X-ON/X-OFF receive handshake pacing is active.

### NOTE

Pressing the [Stop] key will not necessarily result in sending an **XOFF** to the host, the **XOFF** is only sent when the input buffer pointer reaches the upper limit. So, pressing this key is not equivalent to pressing [CTRL] [S] / [CTRL] [Q].

## BELL

The terminal includes a speaker (located in the display monitor) for sounding an audible tone in response to the Bell control code and for alerting the terminal operator when certain error conditions occur (the tone and duration are fixed and cannot be changed).

From the keyboard, you generate the Bell code by simultaneously pressing the [CTRL] and [G] keys.

From a program executing in a host computer, you trigger the bell tone by transmitting a BELL control code (decimal 7).

## WAIT

From a user key or from a program executing in a host computer, you can cause the terminal to pause for approximately 1 second using the following escape sequence:

$E_C@$

Multiple uses of this escape sequence in succession can be used to obtain virtually any desired time delay.

Note that while an  $E_C@$  is in effect, the cursor disappears from the screen, the passing of data from the keyboard to the screen is delayed, and the passing of data from the datacomm buffer to display memory is inhibited.

For example, if you want to sound the bell tone twice in succession with a two-second delay between tones, you could do so using the following control sequence:

< BELL >  $E_C@$   $E_C@$  < BELL >

## MODEM DISCONNECT

You can direct the terminal to “hang up” the modem by sending an  $E_Cf$ . The terminal accomplishes the modem disconnect by lowering the TR/CD (Terminal Ready) line for 2 seconds.



# Alphanumeric Display Control

SECTION

IV

## INTRODUCTION

This section discusses the alphanumeric display control of the terminal.

The display portion of the terminal consists of a display screen and display memory. The display cursor (a blinking underscore mark on the screen) indicates where the next character entered will appear. As you enter characters, the code for the character is recorded at the associated position in display memory, the character is displayed at the cursor position, and the cursor moves to the next character position on the screen. As the screen becomes full, newly entered data causes existing lines to be forced off the screen. Data lines forced off the screen are still maintained in display memory and can subsequently be moved back onto the screen.

You can perform the following display control operations either locally from the keyboard or remotely from a program executing in a host computer:

- Move the cursor up, down, left, or right on the screen.
- Move the displayed data up or down in relation to the current cursor position. When a roll operation forces data off the top or bottom edge of the screen, additional data rolls onto the screen at the opposite edge from display memory.
- Change the content of the screen to the next or previous “page” of data in display memory (a page consists of 24 lines).
- Set or clear a left and right margin.
- Set or clear one or more tab stop positions.
- Move the cursor forward to the next tab stop position or backward to the preceding tab stop position.
- Enable or disable the inverse video, half bright, underline, line blinking and security fields display enhancements.
- Change from one character set to another (e.g. Line Drawing).

In addition, you can do the following screen edit operations either locally or remotely:

- Delete all characters from the current cursor position through the end of display memory.
- Delete the line containing the cursor (subsequent lines are rolled up).
- Change the line containing the cursor to all blanks.
- Turn off screen display except for function key labels and status line.
- Turn off function key labels and status line.
- Delete the character at the current cursor position.
- Insert a blank line immediately preceding (above) the line currently containing the cursor (subsequent lines are rolled down the screen).
- Enable or disable "insert character" mode. When this editing mode is enabled, succeeding characters entered through the keyboard or received from the host computer are inserted to the left of the character at the current cursor position.

#### NOTE:

The HP 2394A terminal can perform the following additional display control and screen edit operations:

- Alter the tab sequence to access unprotected fields in a logical sequence even though they may be grouped vertically.
- Define Unprotected and Transmit Only fields for use in forms mode.
- Enable or disable the all characters, alphabetic-only, auto upshift, alphanumeric, integer-only, signed decimal, implied decimal, constant, integer with zero fill, signed decimal with fill, implied decimal with fill, numeric only, justify, required field, total fill and modified data tag local edit checks.
- Sketch forms, draw lines, and define lines for designing forms.
- Store forms locally for use during a work session in Forms Cache. The Forms Cache Directory allows an application program to determine exactly which forms are currently stored in Forms Cache.

## CURSOR CONTROL

The following topics describe how to alter the cursor/data relationship either manually by using the cursor control keys or programmatically by using escape sequences.

### Home Up ([ $\triangleright$ ])

Pressing the [ $\triangleright$ ] key moves the cursor to the left margin in the top row of the screen and rolls the text in display memory down as far as possible so that the first line in display memory appears in the top row of the screen.

When format mode is enabled, the [▷] key also rolls the text down as far as possible but leaves the cursor positioned at the beginning of the first unprotected field. If no fields have been defined, the cursor will appear at the first column of the first row on the screen.

To perform this function programmatically, use the following escape sequence:

$E_C H$  or  $E_C h$

When memory lock is enabled, the [▷] key rolls the text down as far as possible below the locked area of the screen, instead of below the top of the screen, and leaves the cursor positioned at the beginning of the first unlocked row on the screen. When both format mode and memory lock are active simultaneously, the cursor will go to the first unprotected field on the screen (including the locked area), after rolling all the text down.

#### NOTE

If memory lock is on and the cursor is within the locked area, [▷] will cause the cursor to go down to the first character of the first line of text under the locked area, after rolling the non-locked text down (if required).

### Home Down ([Shift] [▷])

Pressing the [Shift] and [▷] keys moves the cursor to the left margin in the bottom line of the screen and rolls the text in display memory up as far as possible so that the last line in display memory appears immediately above the cursor position.

To perform this function programmatically, use the following escape sequence:

$E_C F$

### Move Cursor Up ([▲])

Each time you press the [▲] key, the cursor moves upward one row in the current column position. If you hold the key down, the cursor movement continues row-by-row until the key is released. When the cursor is in the top row of the screen, pressing this key moves the cursor to the same column position in the bottom row of the screen.

To perform this function programmatically, use the following escape sequence:

$E_C A$

### Move Cursor Down ([▼])

Each time you press the [▼] key, the cursor moves downward one row in the current column position. If you hold the key down, the cursor movement continues row-by-row until the key is released. When the cursor is in the bottom row of the screen, pressing this key moves the cursor to the same position in the top row of the screen.

To perform this function programmatically, use the following escape sequence:

$E_C B$

### **Move Cursor Right ([▶])**

Each time you press the [▶] key, the cursor moves one column to the right in the current screen row. If you hold the key down, the cursor movement continues column-by-column until the key is released.

This function is performed without regard for existing margins. When the cursor reaches the rightmost column of the screen, pressing this key moves the cursor to the leftmost column in the next lower row (from the rightmost column in the bottom row of the screen, the cursor moves to the leftmost column in the top row of the screen).

To perform this function programmatically, use the following escape sequence:

$E_C C$

### **Move Cursor Left ([◀])**

Each time you press the [◀] key, the cursor moves one column to the left in the current screen row. If you hold the key down, the cursor movement continues column-by-column until the key is released.

This function is performed without regard for existing margins. When the cursor reaches the leftmost column of the screen, pressing this key moves the cursor to the rightmost column in the next higher row (from the leftmost column in the top row of the screen, the cursor moves to the rightmost column in the bottom row of the screen).

To perform this function programmatically, use the following escape sequence:

$E_C D$

### **Roll Text Up ([Shift] [▲])**

Each time you press keys [Shift] [▲] (shift cursor up), the non-locked text in display memory rolls up one row on the screen. The top row rolls off the screen, the remaining data rolls up one line on the screen, and a new line of data rolls from display memory into the bottom line of the screen. If you hold both keys down, the text continues to roll upward until you release the keys or until the final line of data in display memory appears in the top row of the screen. In the later case, pressing or continuing to hold down the keys has no further effect. The "roll up" function is illustrated in Figure 4-1 A.

In the configuration and user softkey definition menus, roll up is disabled. In memory lock mode, the unlocked text rolls behind the locked text, as if the bottom line of the locked text is the top of the screen.

To perform this function programmatically, use the following escape sequence:

$E_C S$

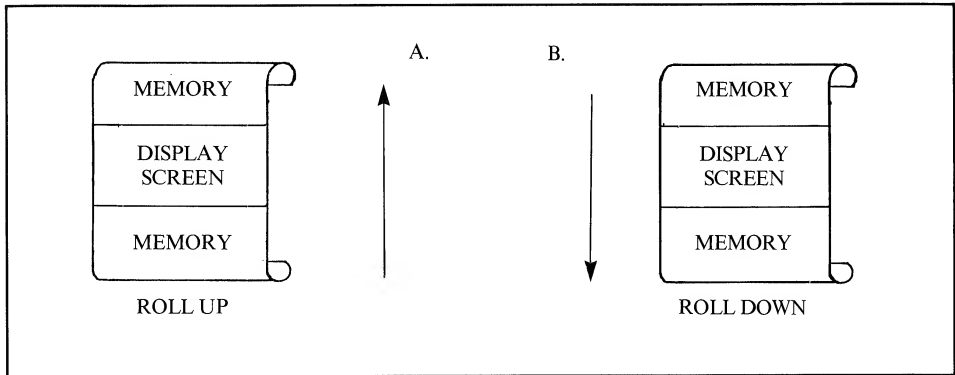


Figure 4-1. The “Roll” Data Functions

### Roll Text Down ([Shift] [▼])

Each time you press keys [Shift] [▼] (shift cursor down), the text in display memory rolls down one row on the screen. The bottom row rolls off the screen, the remaining data rolls down one line on the screen, and a new line of data rolls from display memory into the top line of the screen. If you hold this key down, the text continues to roll downward until you release the key or until the first line of data in display memory appears in the top row of the screen. In the latter case, pressing or continuing to hold down the keys has no further effect. The “roll down” function is illustrated in Figure 4-1B.

In the configuration and user key definition menus, roll down is disabled. In memory lock mode, the unlocked text rolls behind the locked text, as if the bottom line of the locked text is the top of the screen. The cursor position relative to the screen remains unchanged after this operation.

To perform this function programmatically, use the following escape sequence:

$E_C T$

### Next Page/Previous Page ([Next] / [Prev])

The data in display memory can be accessed (displayed on the screen) in blocks that are known as “pages”. A page consists of 24 lines of data. The current page is that sequence of lines which appears on the screen at any given time. The previous page is the preceding 24 lines in display memory. The next page is the succeeding 24 lines in display memory.

The concept of display “pages” is illustrated in Figure 4-2.

Pressing the [Next] key rolls the text in display memory up so that the next page of data replaces the current page on the screen. If you hold the key down, the operation is repeated so that the final line in display memory appears in the top line of the screen.

In the configuration and user-key definition menus, this key is disabled. In memory lock mode, the locked text remains at the top of the screen, and only the unlocked lines are moved.

To perform the “next page” function programmatically, use the following escape sequence:

$\text{E}_\text{C}\text{U}$

Pressing the [Prev] key rolls the text in display memory down so that the previous page of data replaces the current page on the screen. If you hold the key down, the operation is repeated so the first line in display memory appears in the top line of the screen.

In the configuration and user-key definition menus, this key is disabled. In memory lock mode, the locked text remains at the top of the screen, and only the unlocked lines are moved.

To perform the “previous page” function programmatically, use the following escape sequence:

$\text{E}_\text{C}\text{V}$

At the completion of the “next page” or “previous page” function, the cursor is positioned at the left margin in the top line of the screen.

If format mode is on, the cursor will go to the first unprotected field on the new page.

## MEMORY ADDRESSING SCHEME

Display memory positions can be addressed using absolute or relative coordinate values. Display memory is made up of 80 columns (0-79), and typically up to 96 lines (0-95). The amount of memory in the terminal can be determined from byte 0 of the primary terminal status (refer to Section VI). The types of addressing available are:

- Absolute
- Screen Relative
- Cursor relative



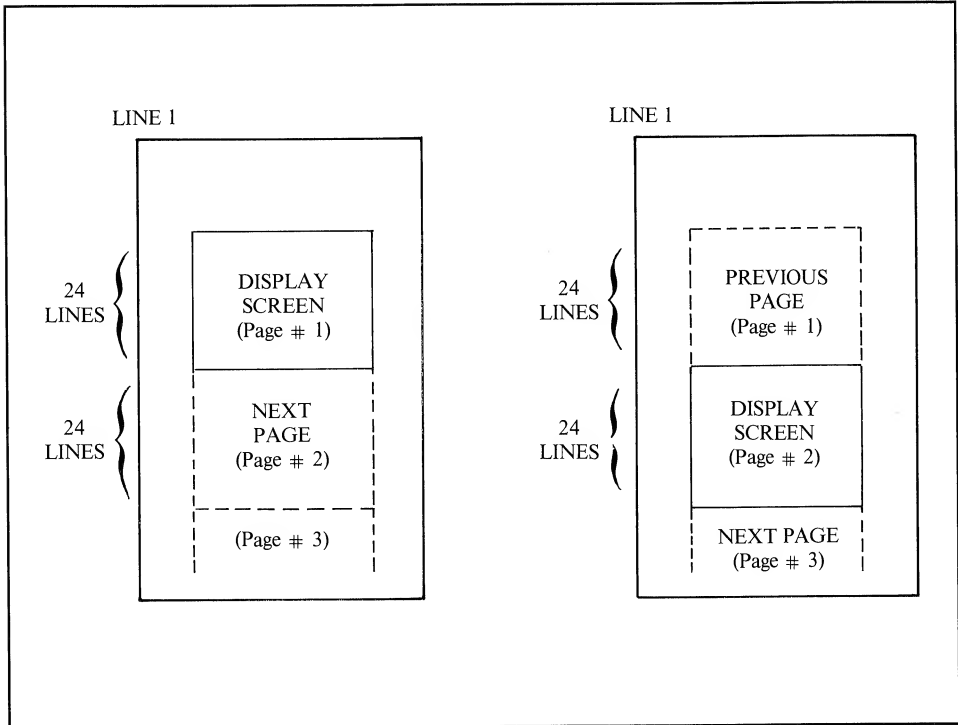


Figure 4-2. Previous Page and Next Page Concepts

**Row addressing.** Figure 4-3 illustrates the way the three types of addressing affect line (or row) numbers. The cursor is shown positioned in the fourth row on the screen. Screen row 0 is currently at row 6 of display memory. In order to reposition the cursor to the first line of the screen the following three destination rows could be used:

- Absolute: row 6
- Screen Relative: row 0
- Cursor relative: row -3

**Column addressing.** Column addressing is accomplished in a manner similar to row addressing. There is no difference between absolute and screen relative column addressing. Figure 4-4 illustrates the difference between absolute and relative addressing. The cursor is shown in column 5.

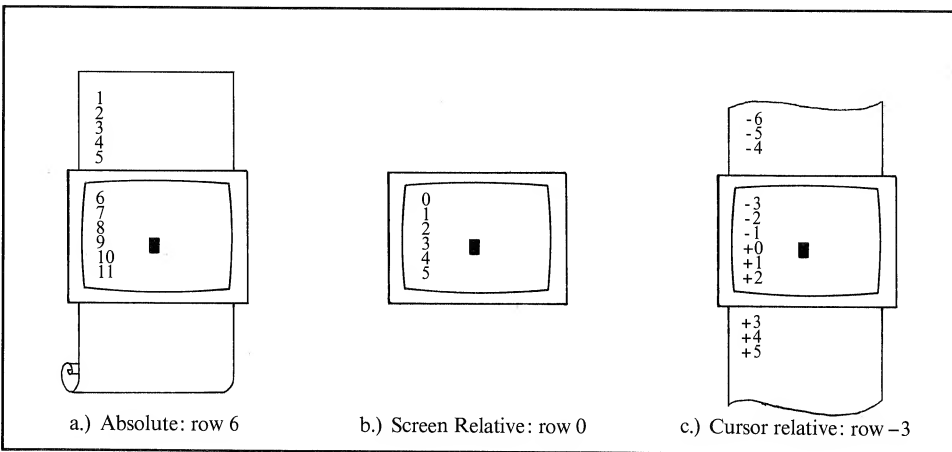


Figure 4-3. Row Addressing

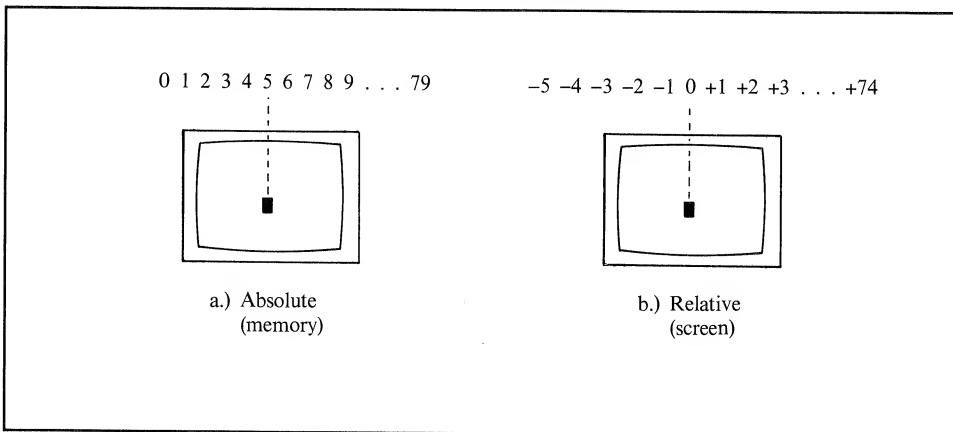


Figure 4-4. Column Addressing

Whenever the row or column addresses exceed those available, the largest possible value is substituted. In screen relative addressing, the cursor cannot be moved to a row position that is not currently displayed (if it is, a roll is performed). For example, in figure 4-3c a cursor relative row address of -10 would cause the cursor to be positioned at the top of the current screen (screen relative row -3). Column positions are limited to the available screen positions (0 to 79 in Figure 4-4a, and -5 to 74 in Figure 4-4b). The cursor cannot be wrapped around from column 0 to 79 by specifying large negative values for relative column positions.

## Cursor Sensing

The current position of the screen cursor can be sensed. The position returned can be the absolute position in display memory or the location relative to the current screen position. (Absolute and relative addresses are discussed under Cursor Addressing).

### Absolute Sensing $E_C a$

Example: The cursor is at column 20, row 40.

computer:  $E_C a$

terminal:  $E_C \&a 020c 040R$

### Relative Sensing $E_C `$

Example: The cursor is again at column 20, row 40, but screen row 0 begins at row 35 of display memory.

computer:  $E_C `$

terminal:  $E_C \&a 020c 005Y$

### NOTE

Columns are numbered 1 to 80 on the screen but 0 to 79 in escape sequences. Similarly, rows are numbered 1 to 24 on the screen but 0 to 23 in escape sequences.

## Cursor Positioning

The cursor can be positioned directly by giving memory or screen coordinates, or by sending the escape codes for any of the keyboard cursor positioning operations.

## Screen Relative Addressing

To move the cursor to any character position on the screen, use any of the following escape sequences:

```

      EC&a <column number> c <row number> Y
or   EC&a <row number> y <column number> C
or   EC&a <column number> C
or   EC&a <row number> Y

```

where:

**<column number>** is a decimal number specifying the screen column (0-79) to which you wish to move the cursor. Zero specifies the leftmost column.

**<row number>** is a decimal number specifying the screen row (0-23) to which you wish to move the cursor. Zero specifies the top row of the screen; 23 specifies the bottom row.

Note that if a screen column > 79 is specified, the cursor stops on column 79 and if a screen row > 23 is specified the cursor stops on row 23.

When using the above escape sequences, the data on the screen always remains unchanged.

If you specify only a <column number>, the cursor remains in the current row. Similarly, if you specify only a <row number>, the cursor remains in the current column.

**Example:** The following escape sequence moves the cursor to the 20th column of the 7th row on the screen:

```
EC&a 6y 19C
```

## Memory Absolute Addressing

You can specify the location of any character within display memory by supplying absolute row and column coordinates. (Note that the row coordinate is ignored when memory lock mode is on). To move the cursor to another character position using absolute addressing, use any of the following escape sequences:

```

      EC&a <column number> c <row number> R
or   EC&a <row number> r <column number> C
      EC&a <column number> C
or   EC&a <row number> R

```

where:

**<column number>** is a decimal number specifying the relative column (0 to 79) to which you wish to move the cursor. A positive number specifies how many columns to the right you wish to move the cursor; a negative number specifies how many columns to the left.

**<row number>** is a decimal number specifying the row coordinate (within display memory) of the character at which you want the cursor positioned. Zero specifies the first (top) row in display memory.

Note that if a screen column > 79 is specified the cursor stops in column 79 and if a row in excess of the screen memory is specified the cursor stops in the last line of the memory.

When using the above escape sequences, the data visible on the screen will (if necessary) be rolled up or down in order to position the cursor at the specified data character. The cursor and data movement will occur as follows:

- If a specified character position lies within the boundaries of the screen, the cursor moves to that position and the data on the screen remains unchanged.
- If the absolute row coordinate is less than that of the top line currently visible on the screen, the cursor moves to the specified column in the top row of the screen and the text rolls downward until the specified row appears in the top line of the screen.
- If the absolute row coordinate exceeds that of the bottom line currently visible on the screen, the cursor moves to the specified column in the bottom row of the screen and the text rolls upward until the specified row appears in the bottom line of the screen.

If you specify only a <column number>, the cursor remains in the current row. Similarly, if you specify only a <row number>, the cursor remains in the current column.

Example: The following escape sequence moves the cursor (and rolls the text if necessary) so that it is positioned at the character residing in the 60th column of the 27th row in display memory:

$\text{E}_C \&a \ 26r \ 59C$

## Cursor Relative Addressing

You can specify the location of any character within display memory by supplying row and column coordinates that are relative to the current cursor position. (Note that the row coordinate is ignored when memory lock mode is on). To move the cursor to another character position using cursor relating addressing, use any of the following escape sequences:

$\text{E}_C \&a \ +/- \ <\text{column number}> \ c \ +/- \ <\text{row number}> \ R$   
 or  $\text{E}_C \&a \ +/- \ <\text{row number}> \ r \ +/- \ <\text{column number}> \ C$   
 or  $\text{E}_C \&a \ +/- \ <\text{column number}> \ C$   
 or  $\text{E}_C \&a \ +/- \ <\text{row number}> \ R$

where:

- <column number> is a decimal number specifying the relative column (0 to 79) to which you wish to move the cursor. A positive number specifies how many columns to the right you wish to move the cursor; a negative number specifies how many columns to the left.
- <row number> is a decimal number specifying the relative row to which you wish to move the cursor. A positive number specifies how many rows downward you wish to move the cursor; a negative number specifies how many rows upward.

When using the preceding escape sequences, the data will (if necessary) be rolled up or down in order to position the cursor at the specified data character. The cursor and data movement will occur as follows:

- If a specified character position lies within the boundaries of the screen, the cursor moves to that position and the data on the screen remains unchanged.
- If the specified cursor relative row precedes the top line currently visible on the screen, the cursor moves to the specified column in the top row of the screen and the text rolls downward until the specified row appears in the top line of the screen.
- If the specified cursor relative row follows the bottom line currently visible on the screen, the cursor moves to the specified column in the bottom row of the screen and the text rolls upward until the specified row appears in the bottom line of the screen.

If you specify only a <column number>, the cursor remains in the current row. Similarly, if you specify only a <row number> the cursor remains in the current column.

Example: The following escape sequence moves the cursor (and rolls the text if necessary) so that it is positioned at the character residing 15 columns to the right and 25 rows above the current cursor position within display memory:

$E_C \&a +15c -25R$

## Combining Absolute And Relative Addressing

You may use a combination of screen relative, absolute and cursor relative addressing within a single escape sequence.

Example: Move the cursor (and roll the text if necessary) so that it is positioned at the character residing in the 70th column of the 18th row below the current cursor position.

$E_C \&a 69c +18R$

Example: Move the cursor so that it is positioned at the character residing 15 columns to the left of the current cursor position in the 4th row currently visible on the screen.

$E_C \&a -15c 3Y$

Example: Move the cursor (and roll the text up or down if necessary) so that it is positioned at the character residing in the 10th column of absolute row 48 in display memory.

$E_C \&a 9c 47R$

## EDIT OPERATIONS

You can edit data displayed on the screen by simply overstriking the old data. In addition, the terminal provides the following edit functions which can be enabled and disabled either manually by using the edit control keys or programmatically by using escape sequences:

- Insert Line.
- Delete Line.
- Insert Character.
- Delete Character.
- Clear Display.
- Clear Line.

(Note that in ANSI/EM52 mode, keys **[Insert line]**, **[Delete line]**, **[Insert character]** and **[Delete character]** are normally disabled, but are available when the ANSI configuration menu is displayed. For details, refer to the reference manual ANSI operation supplement supplied with option 049).

### Insert Line (**[Insert line]**)

When you use the insert line edit function, the text line containing the cursor and all text lines below it roll downward one line, a blank line is inserted in the screen row containing the cursor, and the cursor moves to the left margin of the blank line. Note that when memory lock mode is active, inserting a line within the locked area of the screen does not change the size of the locked area.

From the keyboard, each time you press the **[Insert line]** key, the terminal inserts one blank line. If you hold the key down, the terminal continues to insert blank lines until the key is released.

This edit function is disabled in format mode, and is disabled in the configuration and user softkeys definition menus.

#### NOTE

When display memory is full, inserting a line will cause data to be lost. The first line in display memory will always be the one to be released unless it happens to be on the screen, in which case the last line in display memory will be released.

From a program executing in a host computer, you insert a blank line at the current cursor position using the following escape sequence:

**EcL**

## Delete Line ([Delete line])

When you use the delete line edit function, the text line containing the cursor is deleted from display memory, all text lines below it roll upward one row, and the cursor moves to the left margin. Note that when memory lock mode is active, deleting a line within the locked area does not change the size of the locked area.

From the keyboard, each time you press the [Delete line] key the terminal deletes one line of text. If you hold the key down, the terminal continues to delete text lines until the key is released or until there are no subsequent text lines remaining in display memory. In the latter case, pressing or continuing to hold down this key has no further effect.

This edit function is disabled in format mode, and is disabled in the configuration and user softkeys definition menus.

From a program executing in a host computer, you delete the text line at the current cursor position using the following escape sequence:

$\text{ECM}$

## Insert Character ([Insert char])

When the insert character editing function is enabled, characters entered through the keyboard or received from the host computer are inserted into display memory at the cursor position. Each time a character is inserted, the cursor and all characters from the current position through the right margin move one column to the right. Characters that are forced over the right margin are lost; and when the cursor reaches the right margin, it moves to the left margin in the next lower line and the insert character function continues from that point.

The edit function is meant to be used within that portion of the screen delineated by the left and right margins. If you position the cursor to the left of the left margin, the insert character function works as described above. If you position the cursor beyond the right margin, however, the insert character function affects those characters between the current cursor position and the right boundary of the screen. In such a case, when the cursor reaches the right boundary of the screen, it moves to the left margin in the next lower line and the insert character function continues from that point as described in the first paragraph above.

The movement of existing characters during an “insert character” editing operation is illustrated in Figure 4-5.



When format mode is off, any unprotected, alternate character set, and/or video enhancement fields to the right of the cursor move to the right with the displayable characters. If the cursor is positioned within any such field the insert character function extends the range of the field by one position for each character inserted. Block terminators at or to the right of the cursor position move to the right along with the displayable characters.

When format mode is on and the cursor is positioned within an unprotected field, the insert character function affects only those characters from the cursor position through the end of the current subfield. Block terminators are treated the same as when format mode is off. If the cursor is not within an unprotected field, it automatically moves to the first character position of the next subsequent unprotected field when the first character is inserted.

In the user-keys definition menu, insert character acts the same as in format mode: insert character is disabled in a configuration menu.

From the keyboard, you enable and disable the insert character editing function using the [Insert char] key. When enabled, "Ins Char" is displayed in the status line at the bottom of the screen.

From a program executing in a host computer, you enable and disable the insert character editing function using the following escape sequences:

ENABLE:             $E_CQ$   
 DISABLE:           $E_CR$

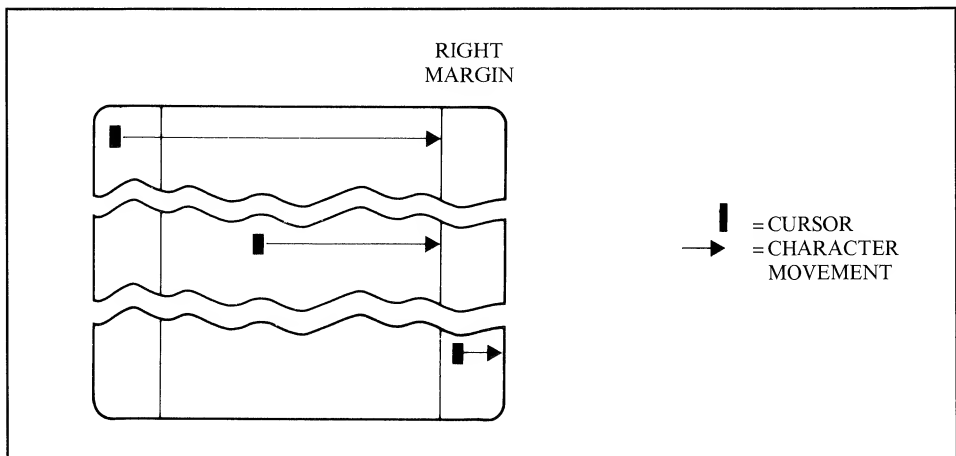


Figure 4-5. Character Insert with Margins

## Delete Character ([Delete char])

When you use the delete character edit function, the cursor remains stationary, the character at the cursor position is deleted, all characters between the cursor and the right margin move left one column, and a blank moves into the line from the right margin.

The edit function is meant to be used within that portion of the screen delineated by the left and right margins. If you position the cursor to the left of the left margin, the delete character function works as described above. If you position the cursor beyond the right margin, however, the delete character function affects those characters from the current cursor position through the right boundary of the screen.

The movement of existing characters during a "delete character" editing operation is illustrated in Figure 4-6.

When format mode is off, any unprotected, alternate character set, and/or video enhancement fields to the right of the cursor move to the left with the displayable characters. If the cursor is positioned within any such field, the delete character function shortens the range of the field by one position for each character deleted. Deleting the first character position of an unprotected field changes the rest of the field to protected. Deleting characters at the start of, or within, a video enhancement and/or alternate character set field does NOT alter the characteristics of the rest of the field. Block terminators to the right of the cursor move to the left along with the displayable characters and are deleted if they are at the cursor position when this function is executed.

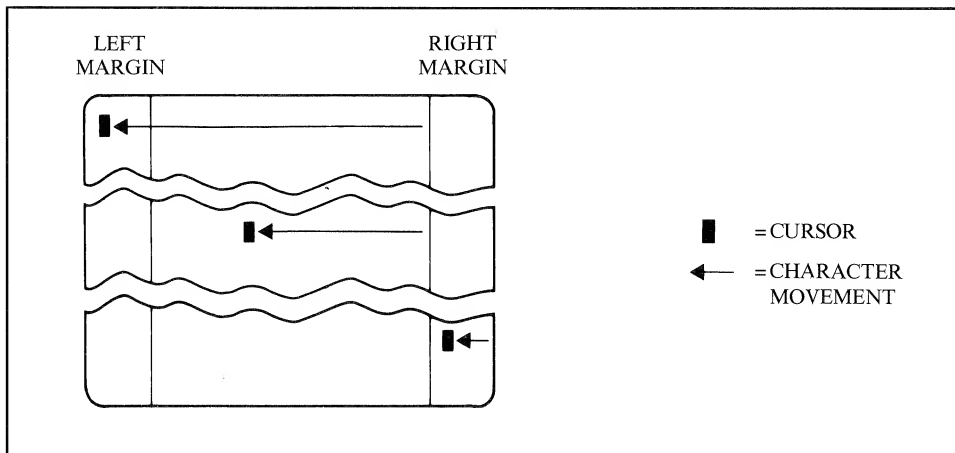


Figure 4-6. Character Delete with Margins

When format mode is on and the cursor is positioned within an unprotected field, this function affects only those characters from the cursor position through the end of the current subfield. If the subfield definition also includes a video enhancement and/or an alternate character set, those characteristics are NOT altered by the delete character function. Block terminators are treated the same as when format mode is off. If the cursor is not within a protected field, the delete character function has no effect.

In the user keys definition menu, delete character acts the same as in format mode: delete character is disabled in a configuration menu.

From the keyboard, each time you press the **[Delete char]** key the terminal deletes one character. If you hold the key down, the terminal continues to delete characters until either the key is released or there are no non-blank characters between the cursor position and the right margin. In the latter case, pressing or continuing to hold down this key has no further effect.

From a program executing in a host computer, you delete the character at the current cursor position using the following escape sequence:

$E_C P$

## **Clear Display ([Clear display])**

When format mode is off, pressing the **[Clear display]** key deletes all displaying and non displaying characters from the current cursor position through the end of display memory.

When format mode is on, pressing the **[Clear display]** key deletes all unprotected displaying and non-displaying characters, all unprotected video enhancements, and any unprotected line drawing characters from the current cursor position through the end of display memory.

This key is disabled in the user-keys definition and configuration menus.

To perform this function programmatically, use the following escape sequence:

$E_C J$

## Clear Line ([Clear line])

When format mode is off, pressing the **[Clear line]** key deletes all displaying and non-displaying characters from the current cursor position through the end of the current line.

When format mode is on and the cursor is positioned within an unprotected field, pressing the **[Clear line]** key deletes all displaying and non-displaying characters and all unprotected video enhancements from the current cursor position through the end of the current field. If the cursor is not within an unprotected field, the **[Clear line]** key has no effect.

In the user-keys definition menu, clear line acts the same as in format mode; clear line is disabled in a configuration menu.

To perform this function programmatically, use the following escape sequence:

$E_C K$

## SETTING AND CLEARING MARGINS

You can redefine the left and/or right margin. These margins affect the cursor positioning for certain functions (such as carriage return, home up, home down, etc.) and establish operational bounds for the insert character and delete character functions. In addition, the left margin is always an implicit tab stop. Data to the left of the left margin or to the right of the right margin is still accessible. Data transfers from display memory to a host computer or to a printer are performed without regard to margins. Format mode, when enabled, clears the margins, creating an 80 character line (1-80).

When you are entering data through the keyboard and the cursor reaches the right margin, it automatically moves to the left margin in the next lower line (note that this operating characteristic can be disabled through the use of the “**InhEolWrp**” terminal configuration parameter; refer to Section II). When you press **[Return]**, the cursor moves to the left margin in the current line if auto line feed mode is disabled or to the left margin in the next lower line if auto line feed mode is enabled. (Note that in ANSI/EM52 mode the margin control keys are disabled.)

When data is being received from a host computer, it enters display memory only within the defined margins. When the cursor reaches the right margin, it automatically moves to the left margin in the next lower line (as mentioned above, this operating characteristic can be disabled through the use of the “**InhEolWrp**” configuration parameter). When  $C_R$  control code is received, the cursor always moves to the left margin in the current line regardless of whether or not auto line feed mode is enabled.

From the keyboard, you set and clear the margins using the **margins/tabs/col** set of function keys. To get to that set, use the following keystroke sequence:

[User System] then 

[f2] margins/ tabs/col
------------------------------

This changes the function key labels to the following:

[f1]	[f2]	[f3]	[f4]	[f5]	[f6]	[f7]	[f8]
START COLUMN	SET TAB	CLEAR TAB	CLR ALL TABS	LEFT MARGIN	RIGHT MARGIN	CLR ALL MARGINS	

To set the left or right margin, move the cursor to the desired column and then press the appropriate function key ([f5] or [f6]). To reset the left margin to column 1 and the right margin to column 80, press [f7].

If you attempt to set either margin incorrectly with relation to the other (e.g., the right margin to the left of the left margin), the terminal rejects it with an audible “beep”.

(Note that in ANSI/EM52 mode, the margin control keys are disabled).

From a program executing in a host computer, you set and clear the margins using the following escape sequences:

SET LEFT MARGIN:  $E_C4$   
 SET RIGHT MARGIN:  $E_C5$   
 CLEAR ALL MARGINS:  $E_C9$

The first two escape sequences set the left and right margin (respectively) at the current cursor position. Before using them, therefore, you will first have to position the cursor at the desired column using one of the cursor control escape sequences described earlier in this section.

## SETTING AND CLEARING TABS

You can define a series of tab stops to which you can move the cursor using the tab and back tab functions (described as separate topics later in this section).

From the keyboard, you set and clear tab stops using the **margins/tabs/col** set of function keys. To get to that set, use the following keystroke sequence:

[User System] then 

[f2]
margins/ tabs/col

This changes the function key labels to the following:

[f1]	[f2]	[f3]	[f4]	[f5]	[f6]	[f7]	[f8]
START COLUMN	SET TAB	CLEAR TAB	CLR ALL TABS	LEFT MARGIN	RIGHT MARGIN	CLR ALL MARGINS	

To set a tab stop, move the cursor to the desired column and then press [f2]. To clear a tab stop, move the cursor to the particular tab stop position and then press [f3]. To clear all existing tab stops, press [f4]. Note that the left margin is always an implicit tab stop and is not affected by [f4]. (Note that in ANSI/EM52 mode, tab stops can also be set using the ANSI configuration menu).

Tab stops that do NOT lie within the area bounded by the left and right margins are ignored when the tab or back tab functions are performed. Also, tab stops are ignored in format mode.

From a program executing in a host computer, you set and clear tab stops using the following escape sequences:

SET TAB:  $\text{E}_C1$   
CLEAR TAB:  $\text{E}_C2$   
CLEAR ALL TABS:  $\text{E}_C3$

The first two escape sequences set and clear (respectively) a tab stop at the current cursor position. Before using them, therefore, you will first have to position the cursor at the desired column using one of the cursor control escape sequences described earlier in this section.

## FORWARD TAB ([Tab▶|])

From the keyboard, you can move the cursor ahead to the next subsequent tab stop using the [Tab▶|] key. In format mode, tab moves the cursor to the beginning of the next unprotected field. At the last field, the cursor wraps around to the beginning of the first field. Tab acts similarly in the user softkeys definition menu and the configuration menu.

From a program executing in a host computer, you can move the cursor ahead to the next tab stop issuing either an  $\text{H}_T$  control code (decimal 9; Control "I") or the following escape sequence:

$\text{E}_C1$

Tab stops that do NOT lie within the area bounded by the left and right margins are ignored by the tab function.

Note that the left margin is treated as a tab stop. When the cursor is positioned at or to the right of the rightmost tab stop, the tab function moves the cursor to the left margin in the next lower line. When the cursor is positioned to the left of the left margin, however, the tab function advances the cursor to the first explicit tab stop in the line (or to the left margin in the next lower line if no explicit tab stops are defined). Note that tabbing the cursor to the next line is the equivalent of a linefeed.

## BACK TAB ([Tab|◀])

From the keyboard you can move the cursor backward to the previous tab stop using the [Shift] and [Tab|◀] keys.

In format mode, configuration menus, and user keys definition menu, the cursor, if within a field, will move to the beginning of the field; otherwise it will move to the first character of the previous unprotected field.

From a program executing in a host computer you can move the cursor backward to the previous tab stop using the following escape sequence:

$$E_{Ci}$$

Tab stops that do not lie within the area bounded by the left and right margins are ignored by the back tab function.

Note that the left margin is treated as a tab stop. When the cursor is positioned at or to the left of the left margin, the back tab function moves the cursor to the rightmost tab stop in the next higher line.

Performing a back tab with the cursor on the left margin of the first row on the screen (or the first unlocked row if memory lock mode is active) is equivalent to performing a roll down.

## FORMAT MODE TAB SEQUENCE CONTROL (HP 2394A ONLY)

The Tab Sequence Control in format mode allows an application to define the tab sequence with respect to unprotected fields.

In format mode, pressing the [TAB] key moves the cursor to the next unprotected field. Normally, the order of unprotected fields is from left to right on a row. When the cursor is positioned at the last unprotected field in a row, pressing the [TAB] key will move the cursor to the next row with an unprotected field.

The HP 2394A allows the tab sequence to be altered. This capability allows, in format mode, tab regions to be created. Within each tab region, the tab key will access unprotected fields with the normal conventions (left to right within a row and top row to bottom row). Regions will be selected with the same convention (left to right and top to bottom). Up to 80 vertical regions may be defined (columns 0-79). These vertical regions may be defined for the complete display memory or for just a range of rows in display memory. If a range of rows is defined, the tab sequence outside the range will occur as normal.

The setting of alternative tab sequences only effects the tab order when using the keyboard [TAB] key or when data entered into the last column of a field causes the cursor to tab to the next field.

The tab sequence escape codes can be used to create a form in one of four different formats.

1. Vertical regions only: All display memory is divided into vertical regions.
2. Top-range: Only a range of rows justified at the top of display memory is divided into vertical regions.
3. Bottom Range: Only a range of rows justified at the bottom of display memory is divided into vertical regions.
4. Mid-range: Only a range of rows in the middle of display memory is divided into vertical regions.

These four formats are discussed below:

### 1. Vertical Regions Only

When in format mode, the normal tab sequence may be altered by splitting display memory into two or more vertical regions up to a maximum of 80 (column 0-79). Each new region is defined by the left most column in which it begins and ends in either the columns before the next defined region or column 79 if another region is not defined.

The escape sequence to define a vertical region is:

`Ec&a<Col 1>s<Col 2>s.....<Col n>S`

where `<Col n>` is the starting column number of the  $n$ th+1 vertical region. Column numbers are specified relative to zero. A region starting in the tenth column would be specified as 9s.

For example, the sequence `Ec&a30s45s` defines three vertical regions (Figure 4-7); the first 30 columns wide, the second 15 columns wide, and the third 35 columns wide. The parameter 0s has special significance and clears all previously set vertical regions. The 0s parameter can be placed at the beginning of an escape sequence to clear all previously set vertical regions (e.g., `Ec&a0s30s45s`).

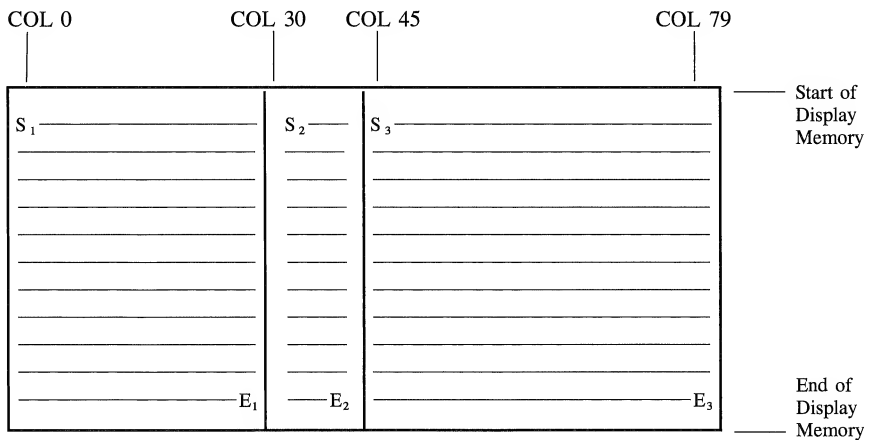


Figure 4-7. Vertical Range Tab Sequence Example `Ec&a30s45s`. ( $S_n$  is the starting point and  $E_n$  is the ending point of the tab sequence in the  $n$ th region.)

Note: Unprotected fields may overlap vertical tab regions. When this occurs, the field belongs to the region in which it begins.



## 2. Top Horizontal Range

Vertical tab regions may be defined for a range of rows beginning at the top of display memory and ending on a specified row. The following escape sequence is used to specify a top justified range of rows:

$\text{Ec\&a} <\text{Col 1}> \text{s} <\text{Col 2}> \text{s} \dots <\text{Col n}> \text{s} <\text{Row 1}> \text{I}$

where  $<\text{Row 1}>$  is the row number of the first row in the second horizontal range. For example,  $\text{Ec\&a30s45s20I}$  defines three vertical regions in the first 20 rows (row 0-19). Rows 20 through the end of display memory retain the normal tab conventions. The parameter 0i has special significance and clears all previously set horizontal ranges of rows. It also can be placed at the beginning of an escape sequence to clear all previous settings.

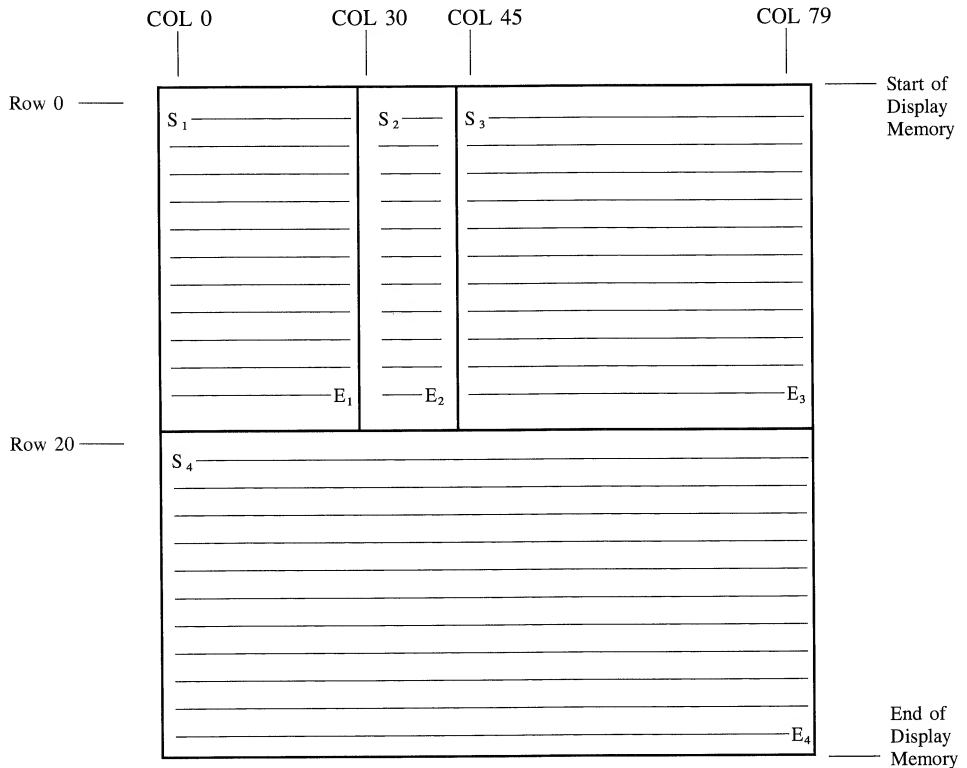


Figure 4-8. Top Horizontal Tab Sequence Example  $\text{Ec\&a30s45s20I}$

### 3. Bottom Horizontal Range

Vertical tab regions may be defined for a range of rows beginning at a specified row and ending at the end of display memory. The following escape sequence is used to specify a bottom justified range of rows:

$EC\&a\langle Row\ 1 \rangle i \langle Col\ 1 \rangle s \langle Col\ 2 \rangle s \dots \langle Col\ n \rangle S$

where  $\langle Row\ 1 \rangle$  is the row number of the first row in the second horizontal range. For example,  $EC\&a\ 20i30s45-S$  defines three vertical regions in the rows beginning with row number 20 and ending at the bottom of display memory. Rows 0-19 retain the normal tab convention.

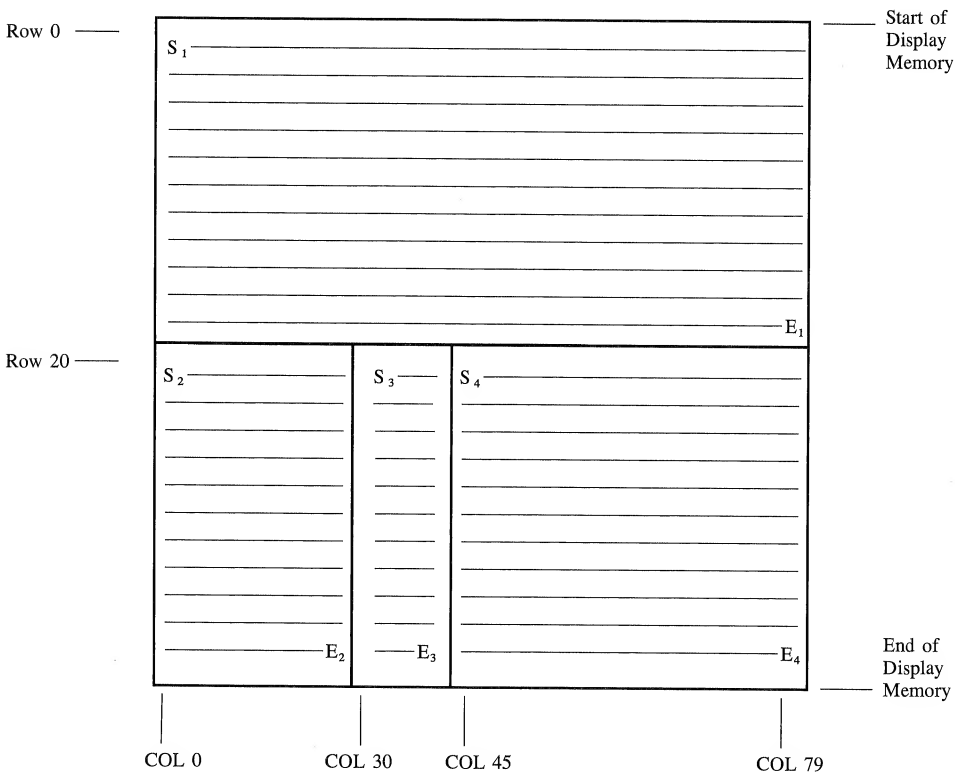


Figure 4-9. Bottom Horizontal Tab Sequence Example  $EC\&a20i30s45S$

4. Horizontal Mid-Range

Vertical tab regions may be defined for a range of rows beginning at a specified row and ending at a specified row. The following escape sequence is used to specify this:

Ec&a<Row 1>i<Col 1>s<Col 2>s.....<Col n>s<Row 2>I

where <Row 1> is the row number of the first row in the second horizontal range and <Row 2> is the row number of the first row in the third horizontal range. For example, Ec&a10i30s45s20I defines three vertical regions in 10 rows (rows 10-19). Rows 0-9 and rows 20 through the end of display memory will retain the normal tab conventions.

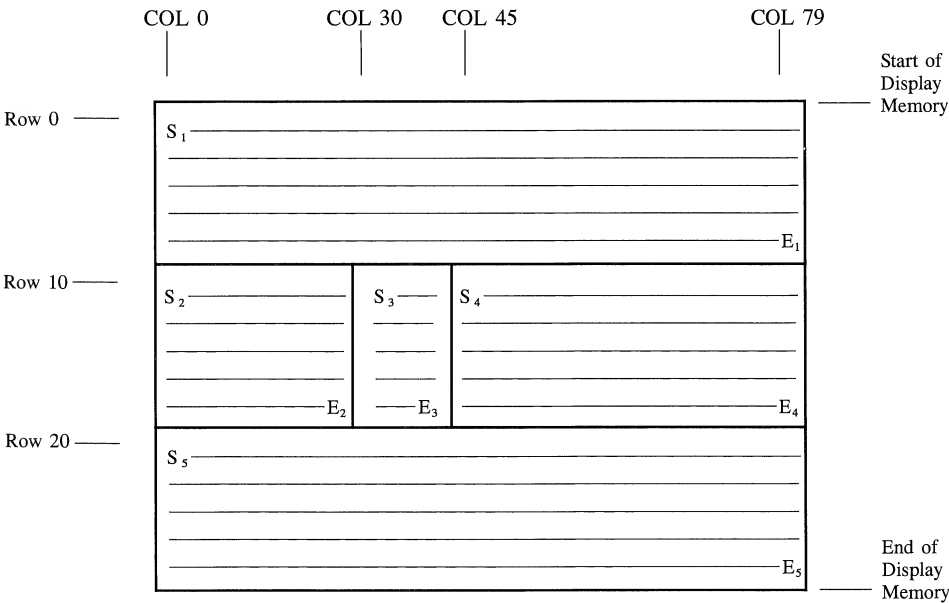


Figure 4-10. Mid-Range Tab Sequence Example Ec&a10i30s45s20I

**Clearing/Resetting Vertical Tab Regions**

When vertical tab regions are defined, they remain in effect until one of the following occurs: 1) hard reset, 2) power off, or 3) tab regions are reset via 0s and/or 0i parameters.

It should be noted that when vertical regions are specified in an escape sequence, they do not clear any previously set regions.

## SCREEN BLANKING

From a program executing in a host computer you can turn off and on the alphanumeric display video, excluding the function key labels and Status Line. This feature may be used to turn off the display video while a form is being drawn into alphanumeric memory. After the form is completed, the video may be turned on. The escape sequences follow:

Turn off display video: `ESC&w13F`

Turn on display video: `ESC&w12F`

This escape sequence does not effect menus.

## DISPLAY ENHANCEMENTS

The terminal includes as a standard feature the following display enhancement capabilities:

- Security Video – characters are not displayed (this enhancement is used in conjunction with fields in which passwords or similar security-sensitive data must be entered through the keyboard).
- Inverse Video – dark characters are displayed against a light background.
- Underline Video – characters are underscored.
- Blink Video – characters blink on and off.
- Half Bright – characters (or background for inverse video) are displayed at half intensity.

You use these enhancements on a field basis. They may be used separately or in any combination. When used, they cause control bits to be set within display memory. If the content of display memory is subsequently transmitted in block mode to a host computer, these control bits are translated into escape sequences which are transmitted along with the displayable text characters.

### NOTE

Using display enhancements reduces the amount of memory available for displaying characters on the screen. Consequently, when the display memory is full activating a display enhancement will cause existing data in memory to be “lost”.

From a program executing in a host computer, or from the keyboard, you enable and disable the various video enhancements by embedding escape sequences within the data. The general form of the escape sequence is as follows:

`ESC&d <enhancement code>`

where enhancement code is one of the uppercase letters A through O and S specifying the desired enhancement(s) or an @ to specify end of enhancement as shown in Table 4-1.

When a video display enhancement is enabled, it affects all the characters from the current cursor position to the end of the line or to an existing change in enhancement (or character set) in the line, whichever occurs first.

Table 4-1. Display Enhancement Character

	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Half-Bright									x	x	x	x	x	x	x	x
Underline					x	x	x	x					x	x	x	x
Inverse Video			x	x			x	x			x	x			x	x
Blinking		x		x		x		x		x		x		x		x
End Enhancement	x															

Note that the escape sequence for “end enhancement” ( $E_C\&d@$ ) or the escape sequence for another video enhancement, will end the previous enhancement.

The terminal also offers a Security field. To enable the security field, the character “s” or “S” is used as an enhancement code.

$E_C\&dS$ : starts security video, and stops the previous video attribute.

$E_C\&ds <A-O>$  enables both security field and the A to O attribute.

$E_C\&ds@$  is converted to  $E_C\&dS$ , and  $E_C\&d@S$  is not executed.

In format mode,  $E_C\&dS$  is not allowed (the bell will ring if it is typed on the keyboard). However, if  $E_C\&dS$  is received before entering format mode it remains in operation (i.e. Security fields cannot be created or removed in format mode).

In format mode,  $E_C\&ds <A-O>$  is converted to  $E_C\&d <A-O>$ .

**Example 1:** Define columns 10 through 14 of line 5 to be inverse video and blinking.

Step 1. Position the cursor at column 10 in line 5.

Step 2. Enter  $E_C\&dC$ .

Step 3. Move the cursor to column 15 in line 5.

Step 4. Enter  $E_C\&d@$  (this ends the enhancements). The field should be light colored.

Step 5. Enter the word TERMINAL beginning in column 9 of line 5. It should appear as shown below. The characters “ERMIN” should be in inverse video and blinking.

10                      15  
 ↓                            ↓  
 T ERMIN AL

You may want to enter some frequently used enhancements into the user keys for ease in entering the enhancements onto the display.

**Example 2:** Enter Underline, Half-bright Inverse, Inverse, and End Enhancement escape sequences into the [f1] through [f4] user keys (Figure 4-11).

Step 1. Press [CTRL] [Menu] to display the menu.

Step 2. Press "NEXT CHOICE" [f2] until "L" is displayed in the attribute field of f1 for local operation.

KEY DEFINITION FIELD	ATTRIBUTE FIELD	LABEL FIELDS	
f1 E <sub>C</sub> &dD	L	UNDER	LINE
f2 E <sub>C</sub> &dJ	L	HALF-BRT	INVERSE
f3 E <sub>C</sub> &dL	L	HALF-BRT	UNDER
f4 E <sub>C</sub> &d@	L	END	ENHANCE
f5 E <sub>C</sub> t	T	f5	
f6 E <sub>C</sub> u	T	f6	
f7 E <sub>C</sub> v	T	f7	
f8 E <sub>C</sub> w	T	f8	

NEXT  
CHOICE

PREVIOUS  
CHOICE

DEFAULT  
VALUES

DISPLAY  
FUNCTNS

[f1]

[f2]

[f3]

[f4]

[f5]

[f6]

[f7]

[f8]

Figure 4-11. Menu After Entering Four Enhancement Escape Sequences


- Step 3. Press **[Tab ►1]** to position the cursor to the label field; then, type a meaningful label to represent the enhancement (e.g. “UNDER” in the first label and “LINE” in the second label field).
- Step 4. Press **[Tab ►1]** to position the cursor to the definition field; then press the “**DISPLAY FUNCTIONS**” key. Now, enter the escape sequence for the enhancement; then, turn off display functions. (Display functions is turned on to enter the “**Ec**”, escape, character).
- Step 5. Tab to the next key fields, and enter the appropriate data in a manner described above.
- Step 6. When you have finished entering the data into the menu, press **[Shift] [User System]** (or **[CTRL] [User System]** or **[Menu]** twice) to return to the normal display with the user key labels.
- Step 7. You may now turn on any enhancement specified in the user keys by positioning the cursor to where you want the enhancement to begin; then, press the appropriate user key. To turn off the enhancement, press the user key containing the end enhancement escape sequence.

## DISPLAY ENHANCEMENTS USING THE SYSTEM FUNCTION KEYS (HP 2394A)

The HP 2394A terminal offers video enhancements accessible through the use of the system function keys (you do **not** have to enter escape sequences through the keyboard.)

From the keyboard, you enable and disable the various video enhancements using the “<enhance video>” set of function keys. To get to that set, use the following keystroke sequence:

**[User System]** **[f5]**



This changes the function key labels to the following:

**[f1]**   **[f2]**   **[f3]**   **[f4]**   **[f5]**   **[f6]**   **[f7]**   **[f8]**

define fields		SET ENHANCMT	SECURITY VIDEO	INVERSE VIDEO	BLINK VIDEO	UNDRLINE VIDEO	HALF BRIGHT
---------------	--	--------------	----------------	---------------	-------------	----------------	-------------

To cause a particular string of text characters to be displayed using one or more of the enhancements, do as follows:

1. Enable the desired enhancement(s) by pressing the associated function key **[f4]**, **[f5]**, **[f6]**, **[f7]**, and/or **[f8]**. When an enhancement is enabled, an asterisk appears in the associated key label.
2. Position the cursor at the first character in the string.
3. Press **[f3]**. The selected enhancements take effect immediately. You will notice that the enhancements begin at the cursor position and continue through the end of the line (or through the next subsequent column in which another display enhancement begins). You will also notice that when you press “<SET ENHANCMT>” **[f3]**, the asterisk automatically disappears from the selected key labels (all enhancements are disabled until you once again explicitly enable them).

4. Position the cursor at the column immediately to the right of the final character in the string.
5. Press [f3]. The enhancements disappear from the cursor position through the end of the line (or through the next subsequent column in which another display enhancement begins). You have actually enabled “no enhancements”, which is recorded in display memory as a distinct control bit pattern that will be translated into an escape sequence ( $E_C\&d@$ ) if the contents of display memory are transmitted to a host computer in block mode.

Moreover, the HP 2392A terminal has the ability to include display enhancements, e.g., half-bright, blinking, etc., within each label field of the user keys menu.

## DESIGNING AND USING FORMS

With the terminal, you can design elaborate data entry forms constructed of varying line types from the optional line drawing set and containing alphanumeric annotations and protected and unprotected fields.

When format mode is enabled, the cursor automatically moves to the start of the first unprotected field in the form. Henceforth, the terminal operator can only enter data into those portions of the display screen which lie within unprotected fields, the remainder of the screen is protected. When the operator enters a character into the last position of a field, the cursor advances to the start of the next unprotected field. In addition, the [Tab ►] and [Shift] [Tab ◀] keys can be used to move the cursor to the start of the next or preceding unprotected field. If the cursor is within a protected field, it automatically advances to the start of the next unprotected field when the operator attempts to type a data character.

You enable and disable format mode programmatically by using the following escape sequences:

ENABLE:  $E_CW$

DISABLE:  $E_CX$

These sequences may be entered directly through the keyboard, executed from a user key definition, or issued from a program in a host computer. The HP 2394A terminal can turn forms mode on and off using the system function keys.

There are three major steps to creating data entry forms:<sup>1</sup>

1. Create the linear structure of the form on the screen using the optional line drawing set.<sup>2</sup>
2. Define the various unprotected fields within the form.
3. Programmatically read the completed form and store it in the host computer for future use.<sup>3</sup>

<sup>1</sup> Also see “Summary (HP 2394A)”.

<sup>2</sup> Or by using the sketch forms capability on the HP 2394A.

<sup>3</sup> See “Forms Cache (Storage) HP 2394A Only”.



## DRAWING FORMS USING THE LINE DRAWING CHARACTER SET

The first step in generating a data entry form is to create the linear structure of the form on the screen along with any constant alphanumeric annotations such as the form's title and the row/column headings. You can do this using the Line Drawing and ROMAN 8 character sets. When you are designing a form through the keyboard, you use  $S_0$  (control-N) and  $S_1$  (control-O) control codes to switch back and forth between the Line Drawing and ROMAN 8 character sets.

The correspondence between the alphanumeric characters and the line drawing elements on the USASCII Keyboard is shown in Figure 4-12 (see Appendix B for national keyboards). Note that as the position of keys generating the same alphanumeric character varies from keyboard to keyboard, so does the position of line elements. Also, national language characters that do not have a corresponding line element generate a space.

You switch from the Base set ROMAN 8 to the Line Drawing set by issuing an  $S_0$  code (control-N) and you switch from the Line Drawing set back to the Base set by issuing an  $S_1$  code (control-O). Note that the  $S_0$  code affects **only** those characters from the current cursor position through the next  $S_0$  or  $S_1$  code, or through the next video enhancement, or through the end of the line, whichever occurs first. Consequently, if the Line Drawing set is enabled at the end of one line on the screen and you also want it enabled at the start of the next line, you will have to explicitly issue another  $S_0$  code at the start of the second line.

Figure 4-13 illustrates a sample form and identifies the keystrokes used for the various different types of line segments.

One approach to generating a form structure through the keyboard is to load two of the user keys with the  $S_0$  and  $S_1$  codes (control-N and control-O, respectively, with displays functions mode enabled), define both as Local keys, define their Labels as "Line Draw" and "Base Set", respectively, and enable them by pressing [Shift] [User System], or [CTRL] [User System], or [Menu] twice. Then draw the form structure and alphanumeric annotations using the Base set. As the form is evolving, use the cursor control keys and the two user keys to switch the linear structure portions of the form to the Line Drawing character set. When doing this, however, be sure that those portions of the form that will be used for data fields are set to the Base set. You may also, if you wish, load some of the more repetitive line definitions into user keys to speed up the drawing of the main body of the form.

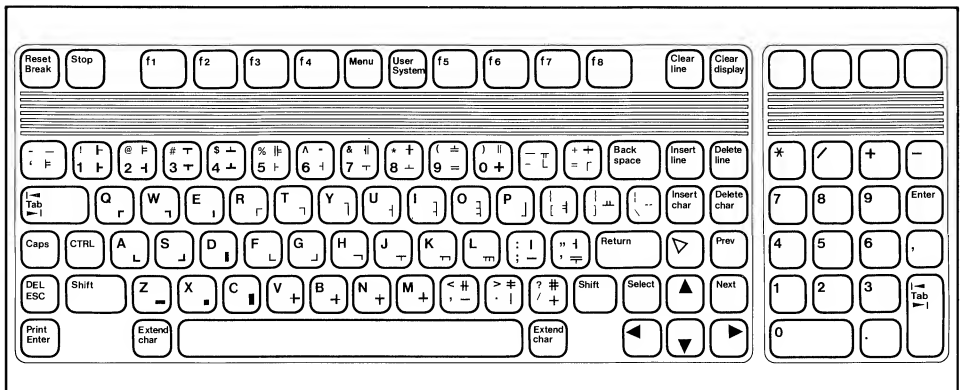


Figure 4-12. Line Drawing Set Elements (USASCII keyboard)

Video enhancements can be incorporated into the form using the user keys. Define the keys as "local", the labels as "half-brt/inverse", "full-brt/inverse", "under/line", "enhance/off", etc. The escape sequences are given previously in this section under "DISPLAY ENHancements". To begin an enhancement, position the cursor on the screen; then, press the user key containing the desired enhancement. Next, to end the enhancement, position the cursor to where you want the enhancement to end (must be in the same line); then, press the key containing the escape sequence for ending the enhancement.

Note: The 2394A provides "ENHANCE VIDEO" softkeys for this purpose.

A program running in the host computer can also be used to draw forms on the terminal display screen.

Note that the Line Drawing set is always the alternative character set (and does not need to be defined). However, the program must go through the same process of shifting out of the base character set ( $S_0$  = control-N = decimal 14) to draw the linear portions of the form, and shift back into the base character set ( $S_1$  = control-O = decimal 15) to structure the alphanumeric portions of the form.

The diagram illustrates a terminal screen with a grid of data entry fields. The screen is surrounded by various function keys. The grid contains several fields with 'XXXXXX' placeholders. The keys include Q, T3, i, DEL, &, W, T7, 2, +, /, ", S, \$, #?, 4, 9, =, >, \*, &, !, 0, 1, ., and Q.

Note: On Swiss French and Swiss German keyboards, in 7-bit operations "è" must be used instead of ">" to generate the "#" element.

Figure 4-13. Sample Data Entry Form

## FORMS MODE (FORMAT MODE)

In forms mode, the terminal prevents you from overwriting or transmitting data in protected fields. Forms mode is normally entered under control of the computer. Forms mode is turned on by sending  $E_CW$  (the cursor is homed to the beginning of the first unprotected field). Normal operation is turned with  $E_CX$  (the cursor remains in its present position). The HP 2394A terminal can turn forms mode on and off using the system function keys (see "Defining Fields From the Keyboard").

### Protected Fields

Fields can be protected so that displayed data cannot be overwritten or sent to a computer. When the terminal is placed in "forms mode" (format mode) all character positions on the screen are protected except those fields that have been specifically defined as "unprotected".

### Unprotected Fields

Data can be written into unprotected fields in the normal manner. Except for the last unprotected field, after reaching the end of an unprotected field, the cursor moves to the beginning of the next unprotected field. For the last unprotected field, the cursor remains at the first character position following the last unprotected field (this is a protected position). Then, if a character is entered from the keyboard, the cursor moves to the first position in the first unprotected field on the screen, and the typed character is entered in this position.

The tab functions can be used to move from one unprotected field to the beginning of the next unprotected field (tab stops are ignored in forms mode).  $E_Ci$  causes the cursor to be positioned at the beginning of the previous unprotected field. Fields are defined as "unprotected" by using  $E_C[$  at the start of the field.  $E_C]$  or the end of the line is used to end the field.

In Figure 4-14, only the fields shown in white are unprotected. Even if the operator moves the cursor to a protected field and types a character, the cursor will move to the nearest unprotected field before displaying the character.

FORM #1876R

Vendor Name		Address		City		State		Zip	
PACIFIC TOOL INC		1273 CRECENT WAY		SAN JOSE		CALIFORNIA		95131	
Quantity		Date		Description		Unit Price		Total Price	
07	16	1976	98	FINISHED STEEL CASTINGS		874738		65.88	
03	19	1976	749	TAPE TRANSPORT BACKPLATES		875483		9753.88	
02	28	1976	13	MILLED FLANGE ASSEMBLY		748563		877.44	
	19								
	19								
H. C. DOUGLAS				04 14 1976					

Figure 4-14. Typical Form

### NOTE

Although the HP 2392A terminal does not support "transmit only" fields, if the "transmit only" escape sequence ( $E_C\}$ ) is sent from the computer, it is redefined as an unprotected field.

## DESIGNING AND USING FORMS ON THE HP 2394A

The HP 2394A includes forms design capabilities that are accessible through the use of TERMINAL function keys (you do **not** have to enter escape sequences through the keyboard).

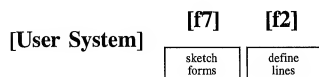
1. Construct the form's structure on the screen using single, double, and bold lines.
2. Define unprotected, protected, and transmit only fields. Each field may be designated with any of the 11 edit checks and may be visually enhanced using the inverse video, underlined video, and blinking video features.
3. Store multiple forms in Forms Cache for later retrieval and use.

### NOTE:

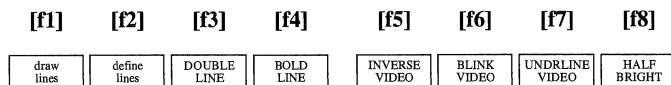
The form must first be sent to the computer and then down loaded to Forms Cache.

## Selecting Line Types

There are three line types available: a single thin line, a double line, and a single bold line. All three are illustrated in figures 4-15 and 4-16. The default line type is the single thin line. To change the line type you must get to the "<define lines>" set of system function keys. One way of doing this is as follows:



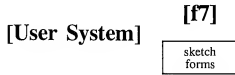
This changes the function key labels to the following:



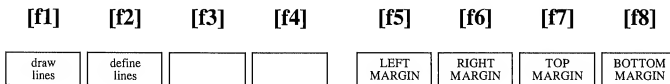
To select a particular line type, press [f2], [f3], or [f4]. The three line types are mutually exclusive. When you select one, the other two types are inactive. When a particular line type is selected, an asterisk appears in the associated key label. The [f5], [f6], and [f7] keys select the desired display enhancement(s) to be used in conjunction with the line drawing set. These function keys act as toggle switches in that they alternately enable and disable the associated video enhancement. When an enhancement is enabled, an asterisk appears in the associated key label.

## Setting Margins

In designing forms you may find it useful to define a top, bottom, left, and right margin and to draw rectangular frames (in relation to either the defined margins or existing alphanumeric text). To do so you must get to the “<sketch forms>” set of system function keys. One way of doing this is as follows:



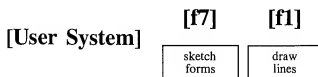
This changes the function key labels to the following:



The [f5], [f6], [f7], and [f8] keys set the left, right, top, and bottom margins (respectively) at the column or row containing the cursor. The left and right margins are the standard margins described earlier in this section. The top and bottom margins, on the other hand, do **not** affect the entry or editing of text through the keyboard nor do they affect the handling of text received from a host computer; they are only used as delimiters in conjunction with the line drawing and framing capabilities.

## Drawing Lines

Once you have selected the desired line type you are ready to begin drawing the linear structure of the form. To do so you must get to the “<draw lines>” set of system function keys. One way of doing this is as follows:



4-36

<b>DEPARTMENT PERSONNEL DATA</b>		
Name:		
Home Address:		
Home Phone: (      )		
Job Title:		
Year Joined Company:	Year Joined Department:	
Salary Curve Code:	Form #AB-1234	

Figure 4-16. Sample Hardcopy Form (For General Office Use)

This changes the function key labels to the following:

sketch forms	ERASE MODE	FROM CURSOR	TO CURSOR	HORIZNTL LINE	VERTICAL LINE	HORIZNTL SEGMENT	VERTICAL SEGMENT
-----------------	---------------	----------------	--------------	------------------	------------------	---------------------	---------------------

The “<sketch forms>” key ([f1]) gets you to the “<sketch forms>” set of system function keys (for setting margins and drawing frames) described in the preceding topic above.

The “<ERASE MODE>” key ([f2]) alternately enables and disables erase mode. When erase mode is enabled, the line drawing function keys erase any existing lines or line segments within their respective operating ranges. When enabled, an asterisk appears in the “<ERASE MODE>” function key label.

The “<FROM CURSOR>” and “<TO CURSOR>” keys ([f3] and [f4]) are used in conjunction with the cursor control keys to draw individual line segments.

The “<TO CURSOR>” key ([f4]) draws a horizontal or vertical line from the current “from” position to the current cursor position. If the two positions are neither in the same row nor the same column, then the terminal draws a rectangular frame with the “from” position in one corner and the current cursor position in the diagonally opposite corner.

Initially the “home up” position is the current “from” position. After you draw a line using any of the line drawing function keys, however, the position of the cursor at the time you pressed the key becomes the new “from” position.

The “<FROM CURSOR>” key ([f3]) defines the current cursor position as the new “from” position. You will use this key when you wish to draw a line segment that is not connected to the previously drawn line or line segment.

The [f5], [f6], [f7], and [f8] “<sketch forms>” function keys operate as described in Table 4-2.



Table 4-2. Line Drawing Function Keys (HP 2394A)

<p>[f5] &lt;HORIZNTL LINE&gt;</p>	<p>This key draws a horizontal line from the left margin to the right margin in the line containing the cursor.</p>
<p>[f6] &lt;VERTICAL LINE&gt;</p>	<p>This key draws a vertical line from the top margin to the bottom margin in the column containing the cursor.</p>
<p>[f7] &lt;HORIZNTL SEGMENT&gt;</p>	<p>This key draws a horizontal line segment between two points in the line containing the cursor. The starting and ending points of the line segment are determined as follows:</p> <ol style="list-style-type: none"> <li>If the cursor is located between two vertical lines (both intersecting the line containing the cursor), then a horizontal line segment is drawn between the two vertical lines.</li> <li>If there is no intersecting vertical line to the right of the cursor but there is to the left, then a horizontal line segment is drawn between the first intersecting vertical line to the left of the cursor and the right margin.</li> <li>If there is no intersecting vertical line to the left of the cursor but there is to the right, then a horizontal line segment is drawn between the left margin and the first intersecting vertical line to the right of the cursor.</li> <li>If there are no intersecting vertical lines to the left or right of the cursor, then a horizontal line is drawn between the left and right margins.</li> </ol>
<p>[f8] &lt;VERTICAL SEGMENT&gt;</p>	<p>This key draws a vertical line segment between two points in the column containing the cursor. The starting and ending points are determined as follows:</p> <ol style="list-style-type: none"> <li>If the cursor is located between two horizontal lines (both intersecting the column containing the cursor), then a vertical line segment is drawn between the two horizontal lines.</li> <li>If there is no intersecting horizontal line above the cursor but there is below, then a vertical line segment is drawn between the top margin and the first intersecting horizontal line below the cursor.</li> <li>If there is no intersecting horizontal line below the cursor but there is above, then a vertical line segment is drawn between the first intersecting horizontal line above the cursor and the bottom margin.</li> <li>If there are no intersecting horizontal lines above or below the cursor, then a vertical line is drawn from the top margin to the bottom margin.</li> </ol>

## NOTE:

The lines and line segments will skip over any existing data characters, unprotected fields, or transmit-only fields.

## Protected and Unprotected Fields (HP 2394A)

See Forms Mode (Format Mode).

### Transmit-Only Fields (HP 2394A Only)

The operator may use the cursor control keys to move the cursor to a transmit-only field and then alter or delete characters within the field. Note that the cursor NEVER automatically moves to a transmit-only field (the [TAB◀] and [TAB▶] keys skip over any transmit-only fields). When the operator presses [ENTER], the data in transmit-only fields is transmitted to the host computer. When a character is entered into the final position of a transmit-only field, the cursor automatically advances to the start of the next unprotected field.

### FIELD ATTRIBUTES (HP 2394A ONLY)

When defining an unprotected or transmit-only field, you must specify a field type (which has a set of implied attributes) and some explicit field attributes. Some of the attributes are checked as each data character is entered into the field, others are checked when the cursor is ready to leave the field, and still others are checked when the data in the form is to be transmitted to the host computer.

The implicit attributes are:

1. The set of characters which may be entered into the field.
2. The format restrictions, if any, for data items entered into the field.
3. Automatic editing, if any, that is performed when the cursor leaves the field.

Table 4-3 summarizes the implicit attributes associated with each field type.

The explicit attributes are:

1. Whether or not the terminal operator MUST enter a data item into the field (<REQUIRED/OPTIONAL>).
2. Whether or not the data entry in the field is to be automatically justified when the cursor leaves the field (<JUSTIFY/NO JUSTIFY>).
3. Whether or not every character position in the field must be filled by the terminal operator (<TOTAL FILL/NO TOTAL FILL>).
4. Whether or not the modify data tag (MDT) for a field is to be set to "on" all the time while in format mode (PERMANENT/REGULAR MDT).

Table 4-3. Attributes by Field Type (HP 2394A)

Field Type	Valid Input Characters	Field Format Restrictions	Automatic Field Editing
<ALL CHARACTERS>	ALL	None	None
<ALPHABETIC>	Upper/lower case alphabetic, space	None	None
<AUTO-UPSHIFT>	ALL (lowercase alphabetic changed to uppercase when they are entered into the field)	None	None
<ALPHANUMERIC>	Upper/lowercase alphabetic, digits, space, period, minus sign, plus sign, comma	None	None
<INTEGER>	Digits, space	No embedded spaces	None
<SIGNED DECIMAL>	Digits, minus sign, plus sign, decimal point, space	No embedded spaces; only one decimal point; plus sign (if present) must immediately precede most significant digit; minus sign (if present) must immediately precede most significant digit or immediately follow least significant digit	None
<IMPLIED DECIMAL>	Digits, plus or minus sign, decimal point, space	No embedded spaces; only one decimal point; plus sign (if present) must immediately precede most significant digit, minus sign (if present) must immediately precede most significant digit or immediately follow least significant digit; completed entry must leave at least one space for decimal point to be added during field edit; if explicit decimal point is entered, the number of digits to the right of it must NOT exceed the number specified by the <Implied Dec Digits> configuration field	Right justified; if there is no explicit decimal point, the implied decimal point is inserted at the proper position

Table 4-3. Attributes by Field Type (Continued)

Field Type	Valid Input Characters	Field Format Restrictions	Automatic Field Editing
<CONSTANT>	None	None	None
<INTEGER FILL>	Digits, spaces	No embedded spaces	Right justified; leading spaces changed to zeros
<SIGNED DECIMAL FILL>	Digits, minus sign, plus sign, decimal point, space	No embedded spaces; only one decimal point; plus sign (if present) must immediately precede most significant digit; minus sign (if present) must immediately precede most significant digit or immediately follow least significant digit	Right justified; leading sign (if present) is left justified; leading spaces changed to zeros
<IMPLIED DECIMAL FILL>	Digits, plus or minus sign, decimal point, space	No embedded spaces; only one decimal point; plus sign (if present) must immediately precede most significant digit; minus sign (if present) must immediately precede most significant digit or immediately follow least significant digit; completed entry must leave at least one space for decimal point to be added during field edit; if explicit decimal point is entered, the number of digits to the right of it must NOT exceed the number specified by the <IMPLIED Dec Digits> configuration field	Right justified; if there is no explicit decimal point, the implied decimal point is inserted at the proper position; leading sign (if present) is left justified; leading spaces changed to zeros
<NUMERIC>	Digits, space, period, comma, minus sign, plus sign	None	None

**Required/Optional Attribute**

Fields may be designated as “required” or “optional”. As shown in Figure 4-17, this attribute is specified by way of the field definition menu. When the terminal operator presses either the <ENTER> key or a user-defined function key that transmits a block of data to the host computer, the terminal checks all “required” fields to be sure that they contain data. If one is found to be empty, then the keyboard is “locked”, the bell is sounded, the cursor moves to the offending field, and an error message appears at the bottom of the screen. To correct the situation, press <RETURN> and then enter suitable data into the field.

**Justify/No Justify Attribute**

For those types of fields which do not include automatic field editing (all characters, alphabetic, auto-upshift, alphanumeric, integer, signed decimal, and numeric), you may select the “<JUSTIFY>” attribute which causes any data within the field to be automatically justified whenever the cursor leaves the field. Like the optional/required attribute, this attribute is specified by way of the field definition menu. For all characters, alphabetic, auto-upshift, and alphanumeric fields, the data is left-justified. For integer, signed decimal, and numeric fields, the data is right-justified. The justify/no justify attribute is ignored for implied decimal, constant, integer fill, signed decimal fill, and implied decimal fill fields.

**Total Fill/No Total Fill Attribute**

For any type of field (except, of course, constant fields) you may specify “< TOTAL FILL >” attribute. This attribute signifies that, if any character position within the field contains a valid character, then every character position within the field must contain a valid character. Like the optional/required and justify/no justify attributes, this attribute is specified by way of the field definition menu. The “< TOTAL FILL >” attribute is checked as the cursor is about to leave the field. If “< TOTAL FILL >” is in effect and all character positions of the field do not contain valid characters, then the keyboard is “locked”, the bell is sounded, the cursor moves to the start of the field, and an error message appears at the bottom of the screen. To correct the situation, press **[RETURN]** and then enter the data item in a manner which ensures that all character positions contain a valid character (right justify it and include leading zeros, for example, for the numeric field types).

### Permanent/Regular MDT Attribute

Data being entered into a field is checked by a modified data tag (MDT). An MDT indicates whether or not data has been entered into the field. There are two types: PERMANENT MDT and REGULAR MDT. If PERMANENT MDT is selected in the field definition menu, all fields created using this definition have their MDTs always set to “on”. This means that all data entered into these fields are recognized as data ready to be transmitted to the computer when the [ENTER] key is pressed or a user-defined function key. If REGULAR MDT is selected, only valid data entered into a selected field sets the MDT for that field to “on”. Usually, all fields in the form are automatically set to “off”.

## DEFINING FIELDS FROM THE KEYBOARD (HP 2394A ONLY)

From the keyboard, you specify the desired field type and explicit attributes using a menu which you access using the following keystroke sequence:

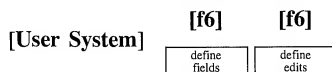


Figure 4-17 illustrates the field definition menu. While the menu is displayed on the screen, the terminal is implicitly in format mode. The menu contains four unprotected fields that you access using the [TAB▶] and [TAB◀] keys. While the cursor is positioned in any of these fields, you select the desired parameters by using the “<NEXT CHOICE> ([f2])” and “<PREVIOUS CHOICE>” ([f3]) function keys. The [f4] function key resets the fields in the menu to their default values.

When you have selected the desired field type and explicit attributes, you then save them by pressing the “<save edits>” ([f1]) function key. When you do this, the menu disappears from the screen and the function key labels change back to the “<define fields>” set, as follows:

[f1]	[f2]	[f3]	[f4]	[f5]	[f6]	[f7]	[f8]
enhance video	START UNPROTECT	START XMITFLD	STOP FIELD	START EDITS	define edits		FORMAT MODE

The <define fields> function keys have the effects described in Table 4-4.

EDIT CHECKS	
FIELD TYPE	<div>0</div> <div>           0. ALL CHARACTERS            1. ALPHABETIC            2. AUTO UPSHIFT            3. ALPHANUMERIC            4. INTEGER            5. SIGNED DECIMAL            6. IMPLIED DECIMAL            7. CONSTANT            8. INTEGER/FILL            9. SIGNED DECIMAL/FILL            10. IMPLIED DECIMAL/FILL            11. NUMERIC         </div>
ATTRIBUTES	<div>OPTIONAL</div> <div>NO JUSTIFY</div> <div>NO TOTAL FILL</div> <div>REGULAR MDI</div>

Figure 4-17. Field Definition Menu (HP 2394A).

Table 4-4. &lt;define fields&gt; Function Keys (HP 2394A)

[f1] <enhance video>	This key redefines the [f1]-[f8] keys to the set of functions which is used for activating the various video enhancements.
[f2] <START UNPROTCT>	This key defines the beginning of an “unprotected” field.
[f3] <START XMITFLD>	This key defines the beginning of a “transmit-only” field.
[f4] <STOP FIELD>	This key defines the end of an “unprotected” or “transmit-only” field.
[f5] <START EDITS>	This key activates the field definition parameters which are currently specified in the “define edits” menu.
[f6] <define edits>	This key displays the field definition menu on the screen (see figure 5-6.).
[f7] <            >	
[f8] <FORMAT MODE>	This key alternately enables and disables format mode. When format mode is enabled, an asterisk appears in the associated screen label.

To define an unprotected field, do the following:

- If you want the field to include edit checks, use the “<define edits>” menu to select the field type and explicit attributes, and then press the “<save edits>” ([f1]) function key.
- Using the cursor control keys, move the cursor to the row and column at which you wish the field to begin.
- If you wish to use any of the video enhancements, press the “<enhance video>” ([f1]) function key, set the desired enhancement(s), and then press the “<define fields>” ([f1]) function key.
- Press the “<START UNPROTCT>” ([f2]) function key.
- If you want the field to include edit checks, press the “<START EDITS>” ([f5]) function key.
- Using the space bar, enter a space for each character that you wish the field to accommodate.
- Press the “<STOP FIELD>” ([f4]) function key.
- If you used any video enhancements, go back to the “<enhance video>” set of function keys and press the “<SET ENHANCMT>” ([f3]) function key (this turns off all enhancements starting at the current cursor position).

To define a transmit-only field, do the following:

- a. If you want the field to include edit checks, use the “<define edits>” menu to select the field type and explicit attributes and then press the “<save edits>” ([f1]) function key.
- b. Using the cursor control keys, move the cursor to the row and column at which you wish the field to begin.
- c. If you wish to use any of the video enhancements, press the “<enhance video>” ([f1]) function key, set the desired enhancement(s), and then press the “<define fields>” ([f1]) function key.
- d. Press the “<START XMIT FLD>” ([f3]) function key.
- e. If you want the field to include edit checks, press the “<START EDITS>” ([f5]) function key.
- f. Type the data that you wish to reside in the field.
- g. Press the “<STOP FIELD>” ([f4]) function key.
- h. If you used any video enhancements, go back to the “<enhance video>” set of function keys and press the “<SET ENHANCMT>” ([f3]) function key (this turns off all enhancements starting at the current cursor position).

If you want to divide a field into subfields, you do so by enabling a new set of field type/attributes. To define a subfield, do the following at the point where you wish the subfield to begin:

- a. Press the “<define edits>” ([f6]) function keys.
- b. Set the menu fields to reflect the desired field type and attributes.
- c. Press the “<SAVE EDITS>” ([f1]) function key.
- d. Press the “<START EDITS>” ([f5]) function key.

You then proceed with the overall field definition as described in the preceding paragraphs above.

Note that any change of video enhancements between the “start field” and “stop field” locations will be cleared whenever they lie within the range of a clear display or clear field operation. If you wish to define a video enhancement for an entire field, you must do so before pressing the “<START UNPROTECT>” or “<START XMIT FLD>” function keys. Video enhancements enabled in conjunction with the start of a subfield (that is, within the overall bounds of a field) will be lost when a clear display or clear field operation is subsequently performed.

### Defining Fields Programmatically (HP 2394A Only)

From a program executing in a host computer, you may define “unprotected” and “transmit-only” fields with the various attributes by using escape sequences. An  $E_C[$  specifies the start of an “unprotected” field, and  $E_C\{$  specifies the start of a “transmit-only” field,  $E_C\&$  sequences define the various attributes of each field or subfield, and  $E_C]$  specifies the end of the field.



The general format of the  $E_C\&e$  sequence is as follows:

$E_C\&e$	[r or R]	(REQUIRED)
	[j or J]	(JUSTIFY)
	[t or T]	(TOTAL FILL)
	[p or P]	(PERMANENT MDT)
	[xxe or xxE]	(Field Type)

where <xx> signifies the desired field type as follows:

- 0 = all characters
- 1 = alphabetic
- 2 = auto-upshift
- 3 = alphanumeric
- 4 = integer
- 5 = signed decimal
- 6 = implied decimal
- 7 = constant
- 8 = integer fill
- 9 = signed decimal fill
- 10 = implied decimal fill
- 11 = numeric

The final identifier in each sequence must be uppercase (<R>, <J>, <T>, <P>, or <E>) and all preceding identifiers must be lowercase (<r>, <j>, <t>, <p>, or <e>).

If the <r>, <j>, <t>, and/or <p> identifiers are omitted from the field definition, the associated attribute is assumed to be "off" (i.e., <OPTIONAL>, <NO JUSTIFY>, <NO TOTAL FILL>, and/or <REGULAR MDT>, respectively). If the field type parameter is omitted from the field definition, it is assumed to be <ALL CHARACTERS>.

The same sequence of operations applies when defining fields and subfields programmatically as when doing so through the keyboard. For example, if you wish the overall field to include video enhancements, you must issue the appropriate  $E_C\&d$  sequence before issuing the  $E_C[$  or  $E_C\{$  sequence. To define the start of a subfield, you issue an appropriate  $E_C\&e$  sequence at the point where the subfield is to begin.

For compatibility with the HP 264x family of terminals, you may also use the following escape sequences to specify field types:

- $E_C6$  = begin alphabetic field (A through Z, a through z, and space only)
- $E_C7$  = begin numeric field (space, 0 through 9, minus sign, plus sign, comma, and decimal point)
- $E_C8$  = begin alphanumeric field (all keyboard characters)

## TRANSMIT MODIFIED FIELDS (HP 2394A Only)

Each field in a formatted display has a Modified Data Tag (MDT) associated with it that indicates whether or not any data has been entered into the field from the keyboard. When the enter key is pressed in block mode, only those fields which have been modified are returned to the host computer.

Normally, when data is entered into a form and the enter key is pressed, the entire contents of the form (all unprotected fields) are returned to the host computer. When the terminal is configured to transmit modified fields only, datacomm traffic can be reduced because only information updated by the terminal operator is returned.

### Setting Modified Data Tags

Whenever format mode gets turned on (enabled), the MDTs for all fields in the form are automatically cleared (set "off"). The entry of any valid characters into a field automatically sets the MDT for that field (set "on"). When one or more fields are cleared through the keyboard (clear display; clear line), the MDTs for the affected fields are set ("on"). When one or more fields are cleared programatically ( $E_{CJ}$  or  $E_{CK}$ ), however, the MDTs for the affected fields are cleared (set "off").

Whenever a field's contents are read by the host computer, its MDTs are cleared (set "off").

### Configuring The Terminal

In the Terminal Configuration menu, there is a field labeled "TRANSMIT" which specifies whether all fields in the form or only those fields which have been modified are to be transmitted to the host computer when the operator initiates a data transfer (using the [ENTER] key, for example). If <TRANSMIT=Modified All Fields>, then only those fields whose MDTs are set "on" are transmitted to the host computer. If <TRANSMIT=Fields>, then all fields in the form are transmitted to the host computer regardless of how their MDTs are set.

The terminal can also be configured with the following escape sequence:

$E_C \&q \langle m \rangle te2 \{ \langle x \rangle Z$

A number inserted in place of the variable <m> identifies the terminal configuration menu. The value of <m> may be a number from 4 to 7. Each number identifies the terminal configuration; however, the number 4 is preferred.

<x> indicates the transmit mode for unprotected fields. If <x>=0 then all fields in the form are transmitted to the host computer regardless of how their MDTs are set. If <x>=1 then only those fields whose MDTs are set "on" are transmitted to the host computer. This escape sequence modifies the terminal configuration in non-volatile memory as well as the active values.

## FORMS CACHE (STORAGE) (HP 2394A ONLY)

On block mode terminals, data entry applications use a form or template to accept each set of information. In applications that require various forms for accepting different sets of data, each form may be displayed many times on the terminal. The time required to display each form on the terminal may add significant overhead to the application.

Forms cache allows an application to pre-load the terminal with information that will subsequently be displayed on the terminal's CRT. From a remote source, forms may be designed and stored in display memory for later retrieval and use. Rather than transmitting a form to the terminal each time it is needed, all of the forms to be used in a work session can be sent to the terminal at the beginning of a session. When a form is needed, it can be transferred from forms cache to the screen using an escape sequence.

### Selecting Forms Cache Memory

The size of forms cache is selected and reserved in 256 byte blocks. This may be accomplished through the terminal configuration menu (see section 10), or from an escape sequence. To select the cache size using the terminal configuration menu the FormsBufSize (256x) field is provided. The number of 256 byte blocks is entered into this field. Forms cache size can also be selected using the following escape sequence:

`ESC & q 4 t e 2 { <number of 256 byte blocks> L`

Available RAM is allocated between display memory and forms cache. If no forms cache is required for an application, enter zero for the number of forms cache blocks. All available memory will be allocated to the terminal display. In all cases, a minimum amount of display memory will be retained to guarantee normal operation of the terminal (see Forms Cache Internals). Minimum display memory will be sufficient to display one page of information in which all rows of data contain video enhancements. If more blocks of forms cache are requested than can be allocated in available memory, the requested amount will be reduced to the maximum that can be successfully allocated. The allocated amount can be verified by issuing a forms cache status request (see Forms Cache Status). Whenever forms cache is re-allocated, current escape sequences are aborted, and record mode is turned off. Video memory is repartitioned when this occurs. The screen is cleared, and any forms residing in cache are purged.

### Defining A Form

Forms are downloaded to cache memory using an escape sequence which may be expressed using any of the following formats:

1. `ESC & p 9 u <form#> p <form-size> L <form-contents>`
2. `ESC & p 9 u <form#> p <<form-contents>> L`
3. `ESC & p 9 u <<form-name>> n <form#> p <form-size> L <form-contents>`
4. `ESC & p 9 u <<form-name>> n <form#> p <<form-contents>> L`

In all cases, form # specifies the user defined number that will be used in subsequent references to the form, by transfer and purge operations. The form number can be a number from 1 to 255. Form numbering does not have to be contiguous, i.e., form numbers can be randomly assigned. A total of 255 different forms may be defined at any one time, however, their total length must not exceed the amount of memory allocated for forms cache.

The form-size parameter may be included in the escape sequence (1 and 3) when the application has determined the length of the form. In this case, form-contents may contain any data including escape sequences, <, and > characters. Exactly form-size characters will be read and stored as the current form. It should be noted that all control characters within the form are stored in cache memory and not immediately acted upon. The following sequence is an example using this format:

```
E_C&p9u50p21LThis is the form text
```

This preceding example stores the 21 bytes of text as form 50 in forms cache.

If the exact length of the form has not been determined, sequence 2 or 4 may be used to store the form. In this case, all characters following the "<" are stored until a ">" character is encountered. If the form contains an embedded right angle bracket or a left angle bracket as part of the form's text, then each angle bracket in the escape sequence must be preceded by a right angle bracket "<". An embedded "<" is interpreted as "take the next character as data". This rule also applies to the form name. The following sequence provides an example of this type of sequence:

```
E_C&p9u50p<This is the text <<including brackets< >> >L
```

Form 50 is stored in forms cache as: This is the text <including brackets>.

Of the four escape sequences that can be used to download forms to cache memory, sequences 3 and 4 provide an additional capability not previously seen before on Hewlett-Packard terminals. With these escape sequences, a form name may be associated with any or all forms downloaded to forms cache. This feature may be used by an application to determine which forms are currently present in cache. The form name provides the ability to define a unique identifier other than 1 to 255 (see Reading the Forms Cache Directory).

A form name must be enclosed by "<>" characters. Any symbol is allowed as part of the name including embedded spaces. An example including the form name is provided below.

```
E_C&p9u<My Form Name>n50p<<This is the text>> >L
```

The preceding form is stored in cache as form 50 with the name: My Form Name. It should be noted that upper and lower case characters in a form name are preserved, e.g., the form name PAYMENTS and the form name payments are treated as two different form names. Control characters, e.g., escape sequences are not allowed in form names. Any such characters occurring in the name definition will be discarded.

If the form number of a downloaded form is identical to a form already stored in cache memory, then the previous form is purged, i.e., the new form replaces the original form. This is also true if the form name is a duplicate. It should be noted that two forms may be purged when a new form is downloaded. This happens when the form number and form name of the new form match two different forms stored in cache. For example, assume that form number 5 has been defined with a form name of ORDERS, and form number 10 has been defined with a form name of SHIPMENTS. If a new form is downloaded with a form number of 10 and a form name of ORDERS, both previous forms will be purged.

## Purging A Form

To purge an existing form from forms cache, one of the following sequences is used:

$E_C \& p9u < \text{form\#} > p0L$

or

$E_C \& p9u < \text{form\#} > pL$

An “S” status is returned at the successful completion of a purge sequence, otherwise an “F” is indicated for a failure. The cache memory used to store the form is reclaimed and will be used when subsequent forms are downloaded. If a form is being redefined, i.e., a form number is being reused, the existing form is automatically purged.

## Transferring A Form To The Screen

Once forms have been defined, they may be copied from cache memory to the screen using the following sequence:

$E_C \& p9u < \text{form\#} > pF$

For example, if form 10 has been previously stored in cache memory, it may be displayed on the screen by sending  $E_C \& p9u10p$  from the host computer. An “S” status is returned at the successful completion of a transfer sequence, otherwise an “F” is indicated for a failure.

The value of forms cache is that information normally displayed many times on the CRT need only be transmitted from the host to terminal one time. Data is displayed much faster from forms cache than over datacomm, and datacomm traffic is reduced.

## Forms Cache Status

A device status request may be used to determine the amount of forms cache currently allocated, how much cache is available for storing additional forms, and whether or not a specific form is currently defined. There are three types of forms cache status escape sequences that can be used:

$E_C \& p9\wedge$

or

$E_C \& p < \text{form\#} > p9\wedge$

or

$E_C \& p < \text{form name} > n9\wedge$

With each sequence, the terminal will respond with a message containing forms cache status information. The format of the response is shown below:

$E_C \backslash p9 < \text{status byte 0} > < \text{status byte 1} > < \text{status byte 2} > < \text{terminator} >$

Each status byte contains status information in the lower 4 bits. The upper 4 bits are set to 0011. This insures that the character code for each byte is always in the range of 30H to 3FH (0-9; < = >?).

Each status byte contains status information in the lower 4 bits. The upper 4 bits are set to 0011. This insures that the character code for each byte is always in the range of 30H to 3FH (0-9; < = >?).

Status byte 0 and 1 contain block count information. Status byte 2 defines the state (present/not present) of a specific form. When a form number or name is included in the status request, the block count is the amount of space remaining to store additional forms within the allocated forms cache memory. Status byte 2 indicates if the form is present (1) or undefined (0).

If a form number is not included in the status request, then status bytes 0 and 1 indicate the number of blocks currently allocated to forms cache. In this case, status byte 2 is always set to 0.

The following example assumes that 10 blocks have been allocated for forms cache. In addition, form 50 has been successfully downloaded to cache memory and occupies 3 blocks. The form name of this form has been defined as RECEIPTS. The current terminator is assumed to be a carriage return (C<sub>R</sub>). For each status request, the response provided by the terminal is shown:

E <sub>C</sub> &p9^	request status of cache
E <sub>C</sub> \p90:O C <sub>R</sub>	terminal response (0: = 10 blocks allocated)
E <sub>C</sub> &p50p9^	request status of form 50
E <sub>C</sub> \p9071 C <sub>R</sub>	terminal response (071 = 7 blocks available, form 50 present)
E <sub>C</sub> &p< RECEIPTS >n9^	request status of form RECEIPTS
E <sub>C</sub> \p9071 C <sub>R</sub>	terminal response (071 = 7 blocks available, form RECEIPTS present)

The terminal response is returned using the handshaking currently defined for the terminal.

## Reading the Forms Cache Directory

The forms cache directory allows an application program to determine exactly which forms are currently stored in forms cache. This provides an application program with the ability to easily determine if any or all of it's associated forms have already been downloaded to cache memory. The feature is useful when several different programs in an application share one or more common forms, or an application is stopped and restarted on the same terminal.

To read the forms cache directory, an extension of the device status request (escape sequence) is used. The sequence is shown below:

E<sub>C</sub>&p< >n9^

This status request will cause the terminal to return a list of the forms currently stored in cache memory. The list will contain form numbers and names (if defined). The format of the list is as follows:

$E_C \backslash p9 < fnum_1 > < < fname_1 > > \dots < terminator >$

The form number ( $< fnum_n >$ ) and form name ( $< < fname_n > >$ ) of each form are returned. If a form has not been assigned a name, the “<” and “>” characters will appear in the list to indicate a null form name. Form numbers and names are returned in the same sequence that they were defined. For example, if form 3 was defined with the name of ORDERS, and form 1 was subsequently defined with no name, the status request and response would appear as follows:

$E_C \& p < > n9 \wedge$	status request from host
$E_C \backslash p93 < ORDERS > 1 < > C_R$	terminal response
	form 3 is defined with name ORDERS
	form 1 is defined without a name

## Forms Cache Internals

Available memory (RAM) is divided between display memory and memory allocated for forms cache. Of the 48K of RAM that exists in the terminal a maximum of 32K is available for display memory. The remaining memory is shared by forms cache and system variables. A minimum of about 12K will always be allocated to forms cache.

When more than 12K is allocated to forms cache, some of the RAM normally used for display memory is reallocated to cache memory. A minimum of 12K of will always be dedicated to display memory. This means that up to 32K of RAM may be allocated to forms cache. Forms are stored in the forms cache in a compressed format and memory management of the space is handled by the terminal firmware.

### Internal Compression

The following items are compressed in the forms cache space:

- Hardware enhancements
- Software enhancements
- $E_C \& a + < offset > C$  sequences
- Repeated spaces and characters

The hardware enhancement escape sequences  $E_C \& d$  and  $E_C \& e$  are compressed into two byte sequences.

The software enhancement escape sequences  $E_C [$  and  $E_C \{$ , are compressed into one byte. When an  $E_C \& e$  (edit check) sequence immediately follows an  $E_C [$  or  $E_C \{$  then they are combined into two software enhancement bytes instead of three. The  $E_C \& a + < offset > C$  sequence is compressed into two bytes.

Strings of 4 or more identical characters are compressed into three bytes. The 1st byte identifies the compressed sequence. The second byte is the repeat count (3-255), and the third byte is the character code. If more than 256 like characters are found, the process is repeated.

Strings of 7 or more printable, i.e., non-control characters are prefixed with a two byte header. The header identifies the string and its length. When a form is copied from cache to the display, the header makes it possible to display strings of text much faster than ordinarily would be possible.

### **Memory Management**

Forms memory is treated as one contiguous space, allocatable through an  $E_C$ &q sequence or the terminal configuration menu. Forms are stored one after the other in forms memory. When a form is purged, all remaining forms move up to fill the vacated space.

Each form is composed of the following internal format. The first byte is the form number (unsigned binary), and the next two bytes contain a link (address) to the next form in cache. If a form name has been defined, it appears next as a variable length field terminated by a unit separator. If no form name has been defined, only the unit separator will be present. The text of the form occupies the remaining space up to the next form number. A dummy form (form 0) is always the last form in cache. This form has a length of zero.



SUMMARY (HP 2394A)

In designing data entry forms for use with a program to be executed in a host computer, you will probably find it useful to take a more systematic approach as follows:

- a. Sketch a draft of the form on a piece of paper to determine the linear structure of the form, the relative location and wording of any desired transmit-only or protected text (such as the form title, column headings, and data field prefixes), and the location and size of all unprotected fields.
- b. Use the “<define lines>”, “<sketch forms>”, and “<draw lines>” sets of system function keys to create the linear structure of the form.
- c. Use the cursor control keys, the “<enhance video>” system function keys, and the alphanumeric keys to create all protected text (such as the form title and column headings).
- d. Use the “<define fields>”, “<define edits>”, and “<enhance video>” system function keys, the cursor control keys, and the alphanumeric keys to create all desired unprotected and transmit-only fields.

Once the form has been designed on the screen you may transfer it to a program executing in a host computer by putting the terminal in block, page mode (make sure format mode is disabled) and then pressing [ENTER].

Figure 4-18 illustrates a typical data entry form (the shaded portions are the “unprotected” fields).

Vendor Name		Address		City	State	Zip
Voucher Date	Units	Purchase and Assembly Details		Post Ref	Cost	
HEWLETT-PACKARD		INITIALED BY:		DATE:		

Figure 4-18. Typical Data Entry Form

C

C

C

# External Printer Control

SECTION

V

## INTRODUCTION

The standard terminal has one port (Port 1), but may be equipped with a Datacomm Module to provide a second port if required (see HP 2392A/HP2394A Accessories Table 1-1).

On terminals equipped with two ports, either of the ports (Port 1 or Port 2) can be used for connecting an external printer. The choice of the printer port is made in the Terminal Configuration Menu (see Section II). With the external printer you may do any of the following:

- Print the line containing the cursor.
- Print all lines from the one containing the cursor through the bottom line on the screen.
- Print all lines from the one containing the cursor through the end of display memory.
- Print only the information contained inside unprotected fields when in Format mode (2394A only).
- Copy a configuration menu from the screen.
- Enable data logging (to occur either from the top or bottom of display memory, as designated by you when you enable it).
- Enable record mode.
- Perform a line feed (advance the paper one line).
- Perform a form feed (advance the paper one page).

All of the above printer control functions can be initiated either locally by operator keystrokes or remotely by escape sequences sent from the host computer.

## Record Mode

Record mode copies data from the datacomm to the selected “to” device(s).

- If in local mode, record mode has no action.
- If in remote mode, the data stream on the datacomm line is sent directly to the selected “to” device (display and/or printer).

Record mode may be initiated from either the keyboard or from an  $\text{E}\&\text{p}$  escape sequence. To initiate record mode from the keyboard, press

[User System] , [f1]  
device  
control , [f1]  
device  
modes , [f2]  
RECORD  
MODE \*

An asterisk will appear in the softkey label to indicate that record mode is enabled. While in record mode, the keyboard is disabled except for the [Shift], [CTRL], [Reset Break], and "RECORD MODE" keys. Pressing [Shift], [Reset Break], or [Shift], [CTRL], [Reset Break], or "RECORD MODE" function key will terminate record mode.

To initiate record mode from an escape sequence, send:

Turn on record mode:  $\text{E}\&\text{p}$  <NUM> p 20C  
(<NUM> p is optional)

The optional <NUM> parameter defines the character which may be used to turn off record mode. <NUM> is the decimal equivalent of an ASCII character that will turn off record mode if it is the first character in a record mode. The default is "0". If "<NUM>" is omitted, or if "0p" is specified, no character will terminate record mode. Termination can only occur by pressing the "RECORD MODE" softkey, pressing [Shift] [Reset Break] (a soft reset), or pressing [CTRL] [Shift] [Reset Break] (a hard reset).

The termination character is valid only for the current activation of record mode (i.e., when record mode is terminated, the termination character returns to "0", the default character).

The terminal returns an "S", "F", or "U" to the host computer if the escape sequence is received from datacomm. An "S" means that the terminal executed the escape sequence successfully; an "F" means that the terminal failed to execute the escape sequence; a "U" means that you terminated execution.

When the status character is sent depends upon whether or not a  $\text{D}_1/\text{D}_2/\text{D}_1$  handshake is enabled. (Refer to the Terminal Configuration Menu discussion in Section II for handshake types). If both the  $\text{D}_1$  and  $\text{D}_1/\text{D}_2/\text{D}_1$  handshakes are disabled (no handshake), the character is sent immediately after the escape sequence is received from the host computer. If the handshake is enabled, the character is sent after record mode is turned off and the appropriate number of  $\text{D}_1$ 's are received from the host computer.

A 256-character buffer is used to hold each record prior to sending it to the specified "to" device(s). If the record exceeds 256 characters, the terminal's handshake holds off any further transmission from the host until the buffer's contents is sent to the "to" device(s). Records shorter than 256 characters are terminated by a  $\text{L}_\text{F}$  (linefeed) character. Again, the terminal's handshake holds off any further transmission from the host until the record in the buffer is sent to the "to" device(s).

If record mode is turned off, the contents of a partially filled buffer will be sent to the "to" device(s).

If the record mode termination character is the first character into the buffer, record mode is terminated; the termination character is not sent to the "to" device(s).

## DATA LOGGING

The terminal includes a mechanism called “data logging” whereby data can be automatically routed to the external printer (provided it has been set as the “to” device). There are two types of data logging: top and bottom.

### Top Logging

When the display memory is full and another line of data is entered through the keyboard or received over a datacomm line, the top line in the display is purged to make room for the new line. With top logging, each line that is purged from the top of the display is sent to the external device. Thus, while the line is “lost” from display memory, it is maintained in a hard copy form. Top logging is disabled if overflow protect is active (see memory lock mode, Section III).

### Bottom Logging

With bottom logging, each time the cursor moves from one line to another as the result of an explicit line feed or an end-of-line-wraparound, the line from which the cursor moved is sent to the external device. This feature allows you to maintain a hard copy “trail” of all lines added to the display in the order in which they were entered and/or received.

When performing data logging in remote mode, the terminal and host computer must be using the ENQ/ACK or XON/XOFF handshakes or they must be using a baud rate that is equal to or less than the rate at which the external device can process data (the device throughput rate).

From the keyboard, you enable and disable data logging using the **LOG TOP** ([f4]) and **LOG BOTTOM** ([f3]) keys. These keys alternately enable and disable top logging and bottom logging, respectively. When either is enabled, an asterisk appears in the associated label.

From a program executing in a host computer, you enable and disable data logging using the following escape sequences:

ENABLE BOTTOM LOGGING:  $\text{E}_\text{C}\&\text{p } 11\text{C}$

ENABLE TOP LOGGING:  $\text{E}_\text{C}\&\text{p } 12\text{C}$

DISABLE LOGGING:  $\text{E}_\text{C}\&\text{p } 13\text{C}$

Both forms of data logging may NOT be enabled simultaneously.

Once either form of data logging is enabled, it remains enabled until explicitly disabled, until the other form of data logging is enabled, until a hard reset is performed, or until the power is turned off.

Note that the keyboard is temporarily locked while a line of data is being “logged”. This may make it difficult to perform any keyboard operations if a large quantity of data is coming into the display over a datacomm line rapidly enough to result in continuous logging.

## DISPLAY TO DEVICE ALPHANUMERIC DATA TRANSFERS

The display is automatically defined as the “from” device in alphanumeric data transfers to the external device. If you define the external printer as destination (“to”) device, you can use the “device control” set of system function keys to print one or more lines of data from the display. One way of going to the “device control” keys is the following keystroke sequence:

[f1]  
[User System], then 

device control
-------------------

This changes the function key labels to the following:

[f1]	[f2]	[f3]	[f4]	[f5]	[f6]	[f7]	[f8]
device modes	TO EXT DEV*	TO DISPLAY*	ADVANCE PAGE	ADVANCE LINE	COPY ALL	COPY PAGE	COPY LINE

To select an external printer as the “to” device, press the “TO EXT DEV” function key ([f2]). An asterisk appears in the associated function key label.

Programmatically, you can define the “to” device by using a device control escape sequence (E<sub>C</sub>&p):

Select external device: E<sub>C</sub>&p 4D

When data is being copied to an external device, keyboard entries are not executed until the transfer is complete (and only the last eight alphanumeric entries are recorded in memory).

### Advance Page ([f4])

When the external printer is selected as a destination device, pressing the **ADVANCE PAGE** key ([f4]) in the **device control** set of system function keys sends an F<sub>7</sub> control code to the printer, thus causing the paper to be advanced to the top of the next page.

Programmatically, you can cause a form feed on the printer by using the following device control escape sequence:

E<sub>C</sub>&p 4u 0C

### Advance Line ([f5])

When the external printer is selected as a destination device, pressing the **ADVANCE LINE** key ([f5]) in the **device control** set of system function keys sends an C<sub>R</sub> I<sub>F</sub> control code sequence to the printer, thus causing the paper to be advanced by one line.

Programmatically, you can cause a line feed on the external printer by using the following device control escape sequence:

E<sub>C</sub>&p 1p 4u 1C

The “p” parameter in the above escape sequences specifies how many line feeds you wish performed. To initiate four successive line feeds, for example, merely substitute “4p” for the “1p” sequence.

## Copy All ([f6])

When the external printer is selected as the destination device, you can copy all lines, starting with the line containing the cursor through the last line of display memory, to the printer. Block terminators are ignored. After each line is printed the cursor moves to the leftmost column in the next lower line (column 1, NOT the left margin). If the cursor is at a line that is beyond the last displayable line, no data is sent to the external printer.

From the keyboard, you copy "all" using the **COPY ALL** ([f6]) key in the **device control** set of system function keys. The transfer can be stopped at the end of the current line by pressing [Return].

From a program executing in a host computer, you copy "all" using one of the following escape sequences:

E\_C&pM or E\_C&p0M

## Format Mode (HP 2394A)

In format mode, the 2394A will either print only the unprotected fields in display memory or all of the display memory depending on the status of the print field in the terminal configuration menu. If <fields> is selected, only unprotected fields are printed and if <all> is selected, the entire display memory is printed.

## Copy Page ([f7])

When the external printer is selected as the destination device, you can copy all lines, starting with the line containing the cursor through the last line visible on the screen, to the printer. Block terminators are ignored. After each line is printed the cursor moves to the leftmost column in the next lower line (column 1, NOT the left margin). If the cursor is at a line that is beyond the last displayable line, no data is sent to the external printer.

From the keyboard, you copy a page of data using the **COPY PAGE** key ([f7]) in the device control set of system function keys. The transfer is stopped at the end of the current line when you press [Return].

From a program operating in a host computer, you copy a page of data using one of the following escape sequences:

E\_C&pF or E\_C&p0F

## Copy Line ([f8])

When the external printer is selected as a destination device, you can copy the line containing the cursor from the display to the printer. The entire line is copied. Block terminators are ignored. After the line is printed, the cursor moves to the leftmost column in the next lower line (column 1, NOT the left margin). If the cursor is on an empty line after the last line of data in display memory, **COPY LINE** should not cause anything to print. If the cursor is in an empty line between two blocks of data, **COPY LINE** causes a C\_R I\_F to be sent to the printer.

From the keyboard, you copy one line of data using the **COPY LINE** key ([f8]) in the **device control** set of system functions keys.

From a program executing in a host computer, you copy one line of data using one of the following escape sequences:

$E_C \& pB$  or  $E_C \& p0B$

## Copy Display Memory ([Print Enter])

When the external printer is selected as a destination ("to") device, you can copy all of display memory to it by using the [Shift] [Print Enter] keys or using an  $E_C0$  sequence. In response to this sequence, the terminal homes the cursor and then copies the entire content of display memory to the printer. Note that when display lock mode is active, cursor home up is below the last locked line.

During the data transfer, block terminators and non-displaying terminators within display memory are ignored.

The  $E_C0$  sequence may be entered through the keyboard, issued from a user-defined function key, or issued from a program executing in a host computer.

When the terminal is in local mode, pressing [Print Enter] key (i.e. enter) or [Shift] [Print Enter] keys (print) performs this same function.

## Copy Menu

When the printer is selected as a destination ("to") device and a configuration menu is currently being displayed on the screen, you can copy the menu to the external printer by pressing the [Print Enter] key by itself or by pressing the [Print Enter] and [Shift] keys together. If the user key definition menu is currently displayed, in local mode you can copy the menu to the external printer using the [Print Enter] key or by pressing the [Print Enter] and [Shift] keys together. In remote mode, you can copy the menu by pressing the [Print Enter] and [Shift] keys together. Using the [Print Enter] key transmits the function key definition escape sequence  $E_C \& f \dots$  to the computer (see Section III).

## Device Control Completion Codes

After issuing a copy line, copy page, copy all (from line containing cursor to end of memory), advance line, or advance page  $E_C \& p$  sequence, the remote program determines whether or not the operation was successfully performed by reading a one character status byte from the terminal. The transfer is initiated by a handshake with the terminal. For example, from a BASIC program, an INPUT statement would be used. The terminal responds by sending an "S", "F", or "U". An "S" indicates successful completion, an "F" indicates that the operation failed, and a "U" indicates that the terminal operator interrupted the data transfer by pressing [Return]. Note that these completion



codes cannot be suppressed by configuration parameters or any other means. They are always transmitted and your programs should include input commands explicitly for accepting them. The keyboard is disabled (“locked”) until the status is sent.

Note that in either character or block line mode, the terminal sends a  $C_R$  (or a  $C_R L_F$  if auto line feed mode is enabled) following the completion code. In block page mode, it sends a block terminator character (as defined in the terminal configuration menu described in Section II of this manual).

In addition to completion codes, you can send a device status request to the terminal following the print operation (refer to “Device Status,” Section VII).

## COMPUTER TO DEVICE DATA TRANSFERS

When the external printer is selected as a destination (“to”) device, you can initiate a data transfer from a program executing in a host computer to the printer by using the following device control escape sequence:

$E_C \& p$  <character-count> W <record>

where:

<character-count> is an integer within the range 1-256 specifying the number of binary bytes in <record>. The record is terminated when the specified number of binary bytes have been transmitted. If this parameter is not present, normal (ROMAN 8) transfers are initiated and the record is terminated when the 256th data byte after the “w” is transmitted or by the first  $L_F$  code, whichever occurs first. If the record is terminated by an  $L_F$ , the  $L_F$  is also passed to the printer.

<record> is the data record to be transmitted.

Example: Send the next 15 binary bytes from the computer to all “to” devices.

$E_C \& p$  15W

This escape sequence is recognized only when received over a datacomm line. It is ignored if entered through the keyboard.

You may include the desired destination device assignment(s) within the escape sequence by using the “d” command parameter. You may also, prior to issuing the above escape sequence, define the desired destination devices either locally through the keyboard or programmatically by way of a separate device control (Fc&p) sequence. In any case, the only destination devices that are recognized by this type of data transfer operation are the display (3d) and external printer (4d).

If no destination devices are specified within the above escape sequence, the current “to” device assignments are used. If nothing is currently selected as a “to” device, the data record is accepted over the host computer port and then is discarded by the terminal (also an “F” is returned as the device control competition code).

Binary transfers (where parity checking is disabled) are of the form Fc&p <character-count> W <record>; all eight bits received are passed to the destination devices. Other transfers (USASCII, Extended Roman), with parity checking performed as set in datacomm configuration menu, are of the form Fc&pW <record>, where Lf or the 256th data character terminates the record.

1. If the escape sequences does NOT include a <character-count>, the following applies:
  - If EnqAck = YES in the active data communications configuration menu, the datacomm firmware strips all ENQ characters from the incoming data and responds to each by transmitting an ACK.
  - If EnqAck = NO in the active data communications configuration menu, all incoming characters (including ENQ and ACK but excluding NULLS and DELS) are treated as data bytes. NULLS and DELS are stripped from the incoming data.
2. If the escape sequence includes a <character-count>, then the following apply:
  - If EnqAck = YES in the active data communications configuration menu, the first ENQ character received in the following data stream is treated as part of the ENQ/ACK handshake (the ENQ is stripped and an ACK returned). Only after returning ACK does the terminal enter binary mode. All subsequent characters (including ENQ, ACK, NUL and DEL) are treated as 8-bit data bytes, regardless of the datacomm configuration menu. (All characters preceding the ACK are processed according to the datacomm configuration menu).
  - If EnqAck = NO in the active data communications configuration menu, binary mode is active immediately after the “W” character of the escape sequence. All ENQ, ACK, NULL and DEL characters are treated as data.

When transferring a data record from the host computer to the printer using the above device control escape sequence, the remote program determines whether or not the operation was successfully performed by reading a one character status byte from the terminal. The transfer is initiated by a handshake with the terminal. For example, from a BASIC program, an INPUT statement would be used. The terminal responds by sending an "S", "F", or "U". An "S" indicates successful completion, an "F" indicates that the operation failed, and a "U" indicates that the terminal operator interrupted the data transfer by pressing [Return]. Note that these completion codes cannot be suppressed by configuration parameters or any other means. They are always transmitted and your programs should include input commands explicitly for accepting them. The keyboard is disabled ("locked") until the status is sent.

Note that in either character or block line mode, the terminal sends a CR (or a CR LF if auto line feed mode is enabled) following the completion code; in block page mode, it sends a block terminator character (as defined in the terminal configuration menu).

In addition to completion codes, you can send a device status request to the terminal following the print operation (refer to "Device Status," Section VII).

If a data comm error occurs during the transmission of the data record, the device control completion code is unpredictable. Data comm errors are reported by way of the terminal status bytes described in Section VII of this manual.

## CONFIGURING THE EXTERNAL PRINTER

Port 1 is always a serial RS232C/HP422 port. On terminals equipped with two ports, Port 2 can be serial or parallel (Centronics). Configuring an external printer consists of assigning it to the port and specifying parameters in the external device configuration menu.

### Printer Cabling

The available printer cables are listed in Table 5-1.

The male connector on the end of the cable attaches to the external device port on the rear of the terminal, and the other end attaches to the printer as shown in Figure 5-1.

You can use cables other than those listed in Table 5-1, as long as they are equipped with the correct connector and their pinouts are compatible with the HP cables.

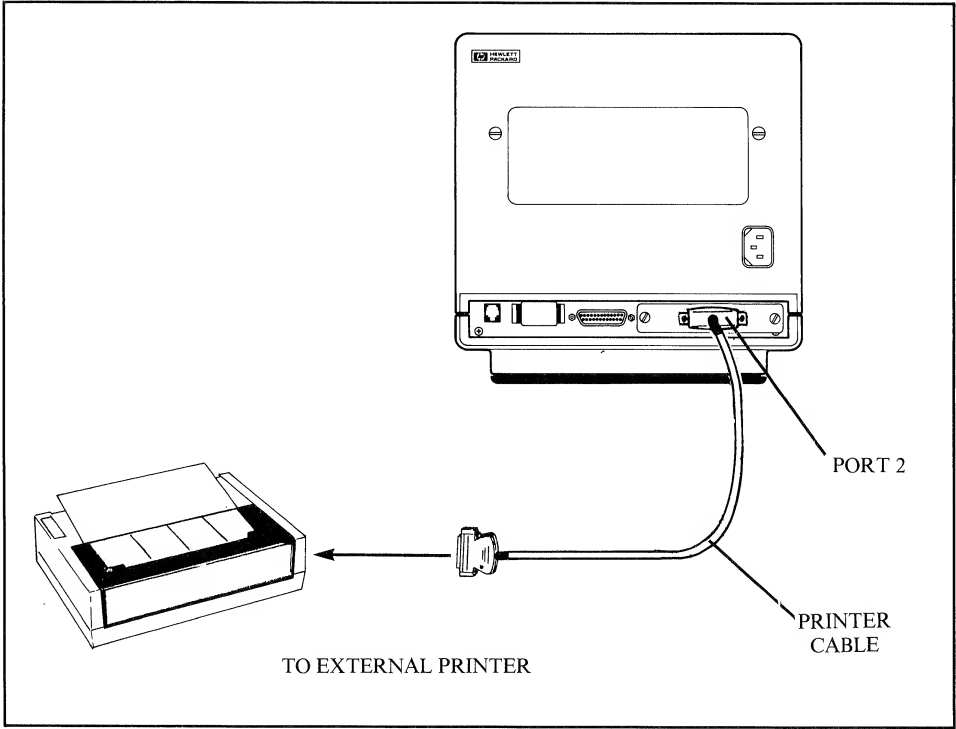


Figure 5-1. External Device Port Cabling

Table 5-1. External Device Port Data Communications Cables

Cable No.	HP Part No.	Description
<b>Serial Printer Cable</b>		
40242G	—	RS232C PRINTER CABLE MALE 25 PIN TO MALE 25 PIN Male RS232C 25-pin connector for interfacing the terminal to RS232C compatible printers such as the HP 263X, HP 267X, HP 293X. Length: 5 meters (16 feet)
<b>Parallel Printer Cable</b>		
40242D	—	PRINTER CABLE AMPHENOL MALE 36 PIN TO MALE 36 PIN Length: 2 meters (6.6 feet)

## Filling In The Configuration Menu

Now that you have made the physical connections between the terminal and the external printer, you are ready to configure the terminal's external device port.

To configure the port, first use the following keystroke sequence:

[User System], 

config keys
----------------

  
[f8]

This changes the function key labels to the following:

		datacomm config	ext dev config	terminal config			
[f1]	[f2]	[f3]	[f4]	[f5]	[f6]	[f7]	[f8]

The **ext dev config** function key, when pressed, causes the external device configuration menu to appear on the screen and redefines the function keys to a set of functions that will assist you in manipulating the various parameters within the menu (see Figure 5-2). The configuration menu displays the currently stored menu parameters from non-volatile memory.

### NOTE

Whenever a configuration menu is on the screen, incoming data, including escape sequences, is stored in buffer (up to 256 bytes) and not processed and no data is transmitted, until the menu is exited. (Refer to Section VI for pacing considerations).

Whenever a configuration menu is displayed on the screen, the terminal is implicitly in format mode. The menu contains a set of unprotected fields that you access using the [Tab◀▶] key. For most of the fields (the ones containing the underlined video enhancement) you select the desired parameters using the “NEXT CHOICE” ([f2]) and “PREVIOUS CHOICE” ([f3]) function keys.

The meanings of the various fields are described in Table 5-2.

EXTERNAL DEVICE CONFIGURATION							
BaudRate	2400	Parity/Data Bits	None/8	PrinterNulls	000		
Printer Type	ROMAN 8						
XmitPace	None	SRRXmit	NO				
		SRRInvert	NO	CS(CB) Xmit	NO		
HP 2392A:							
[f1] SAVE CONFIG	[f2] NEXT CHOICE	[f3] PREVIOUS CHOICE	[f4] DEFAULT VALUES	[f5]	[f6]	[f7] DISPLAY FUNCTNS	[f8] config keys
HP 2394A:							
[f1] SAVE CONFIG	[f2] NEXT CHOICE	[f3] PREVIOUS CHOICE	[f4] DEFAULT VALUES	[f5] POWER ON VALUES	[f6] ACTIVE VALUES	[f7] DISPLAY FUNCTNS	[f8] config keys

Figure 5-2. External Device Configuration Menu

#### NOTE

When using record mode or pass through mode ( $E_C \& pW$ ) to transfer data directly from the datacomm to the printer, both the external device and the datacomm should use identical data bits, also the same method of accessing extended Roman characters as specified in Table B-5 should be used. If not (or not possible), unexpected results may occur.

When the external printer is a parallel printer (or current loop), the following configuration fields should be set to NO: SRRXmit and CS(CB)Xmit.

The HP 2392A/HP 2394A Service Manual (part number 02394-90003) lists the configurations recommended for different types of printer.

Table 5-2. External Device Configuration Menu Fields

<b>BaudRate</b>	<p>This field specifies at what speed you want the data transmission to take place (in bits per second).</p> <p>Values:    <b>110</b>            <b>600</b>            <b>2400</b>                   <b>134.5</b>        <b>1200</b>        <b>4800</b>                   <b>150</b>            <b>1800</b>        <b>9600</b>                   <b>300</b>                    <b>19200</b></p> <p>Default:   <b>2400</b></p>
<b>Parity/Data Bits</b>	<p>This field specifies what type of parity generation and checking you wish used with each data character. (Note that parity must be set to None for binary transfers) and the number of bits per character transmitted or received.</p> <p>Values:    <b>None/8</b>        (no parity bit)                   <b>0's/7</b>            (parity bit always zero)                   <b>ODD/7</b>            (odd parity)                   <b>1's/7</b>            (parity bit always one)                   <b>EVEN/7</b>        (even parity)</p> <p>Default:   <b>None/8</b></p>
<b>PrinterNulls</b>	<p>This field specifies the number of null codes (0-255) to be transmitted to an external printer after each ASCII control code.</p> <p>Value:     <b>0-255</b></p> <p>Default:   <b>0</b></p>
<b>PrinterType</b>	<p>This field specifies which type of language control should be used to send national characters to the external printer. There are two types of printers: those which support the Roman Extension set as a secondary set, accessed either by the SO control code or by the 8th bit (extended Roman) and those which allow the ROMAN 8 set to be accessed with full 8-bit control (the ROMAN 8 is the base set): ROMAN 8.</p> <p>Value:     <b>EXT ROMAN, ROMAN 8</b></p> <p>Default:   <b>ROMAN 8</b></p> <p>See Appendix B for more details.</p>

Continued...

Table 5-2. External Device Configuration Menu Fields (Continued)

<b>XmitPace</b>	<p>Transmit pacing is a mechanism by which the remote device can control (stop and resume) the transmission of data from the terminal.</p> <p>If enabled, transmit pacing is performed using XON and XOFF control codes. When the terminal receives an XOFF code (&lt;DC3&gt;), it stops transmitting data. When the terminal subsequently receives an XON code (&lt;DC1&gt;), it resumes transmitting data.</p> <p>If this field is set to "NONE", the terminal does NOT recognize the &lt;DC1&gt; and &lt;DC3&gt; codes as XON and XOFF.</p> <p>For other forms of transmit pacing, refer to the descriptions of the SRRXmit and CS(CB)Xmit fields below.</p> <p>Values:   NONE           XonXoff</p> <p>Default:  NONE</p>
<b>SRRXmit</b>	<p>This field specifies whether or not an "on" state (+ 12 V) on the RS232C Secondary Receiver Ready (SRR) or Secondary Carrier detect (SCD) control line is a required condition for transmitting data. This mechanism is primarily used in conjunction with printers which must be able to control the transmission of data from other devices. The SRR/SCF control line is connected to RS232C pin number 12.</p> <p>This field should be set to NO when using a parallel printer.</p> <p>Values:   YES           NO</p> <p>Default:  NO</p>
<b>SRRInvert</b>	<p>This field applies only when the SRRXmit field is set to "YES". When both the SRRXmit and SRRInvert are set to "YES", the true state of the RS232C Secondary Receiver Ready (SRR) or Secondary Carrier Detect (SCF) control line is inverted from + 12 V to - 12 V.</p> <p>Values:   YES           NO</p> <p>Default:  NO</p>
<b>CS(CB)Xmit</b>	<p>This field specifies whether or not an "on" state (+ 12 V) on the RS232C Clear to Send (CS/CB) control line is a required condition for transmitting data.</p> <p>This field should be set to NO when using a parallel printer.</p> <p>Values:   YES           NO</p> <p>Default:  NO</p>



# Data Communications

SECTION

VI

## INTRODUCTION

The standard terminal has one port (Port 1), a combined RS232C/HP422 port. The terminal can be optionally equipped with one of several interfacing modules (see Table I-1) to provide a second port. Either Port 1 or Port 2 may be used to connect the terminal to the host computer (Port 2 cannot be used when equipped with a parallel port); the port used for the computer connection being selected in the terminal configuration menu (see Section II).

The term “data communications” (or “datacomm”) refers to the transfer of data between the terminal and a host computer.

There are several ways to connect the terminal to a computer. To arrive at a particular way you must compare a number of factors and make a series of decisions. After selecting the necessary equipment and cables, you must then physically connect the terminal to the computer (or to the modem, if that is what you have chosen) and configure the terminal for use with the particular type of data communications link.

This section is divided into four parts:

1. The first is a general discussion that should help you decide what type of equipment and cabling you need for the data link.
2. The second tells you how to physically install the terminal.
3. The third tells you how to configure the terminal to operate properly with the selected type of data link.
4. The final part provides programming reference material for someone who is writing a datacomm driver or controller program to communicate with the terminal in a point-to-point environment.

Before proceeding with the decision making process, it may help to briefly define the most important terminology as it pertains to data communications.

Communications Link:	The means by which a terminal is connected to a host computer. This always includes some type of communications line (a coaxial cable, the public telephone network, or a leased telephone line), and it may also include a pair of modems (one at each end of the line).
Point-to-Point:	A data communications configuration in which a single terminal is connected to a host computer over a communications link. The terminal is designed for use with a point-to-point communications link.
Asynchronous:	A mode of transmission in which each data character is framed by a “start bit” and one or more “stop bits”. The interval between successive data characters is random. The terminal is designed for use with an asynchronous communications link.
Full Duplex:	A communications link in which data can be transmitted in both directions simultaneously. The terminal is designed for full duplex data communications.
Character Mode:	When the terminal is operating in character mode, it sends data characters to the computer one at a time as they are typed into the keyboard.
Block Mode:	When the terminal is operating in block mode, data characters typed on the keyboard are merely stored in display memory. When a block transfer is subsequently initiated (by the host computer or by pressing the <b>[Enter]</b> key), a group of data characters is sent from the terminal to the computer as a block.

A point-to-point configuration is the standard form of data communications within the industry (it is sometimes referred to as a “Teletype-compatible” communications link). Point-to-point is supported by most computers. At any given time, it accommodates only one terminal per communications link; it may, however, operate in either character mode or block mode.

A major consideration in selecting which type of connection to use is the anticipated distance between the terminal and the computer. If the terminal will be located in the vicinity of the computer system, you may use a hardwired connection. The Electrical Industry Association (EIA) Standard RS232C (to which the terminal conforms) limits cable lengths to a maximum of 15 meters (50 feet).

HP422 is limited to a maximum of 1200 meters (4000 feet).

A current loop is limited to a maximum of 600 m (2000 ft).

Another consideration is the desired availability of the particular computer port. If you wish to have it available (at different times) to terminals in diverse and/or varying locations, then you should choose a modem connection with dial-up capability.

## HARDWIRED CONNECTIONS

If you decide on a point-to-point hardwired connection, the only decision that remains to be made is the type of cable to be used. The available cables are summarized in Table 6-1. Please note that a hardwired connection for your terminal is always full duplex (the terminal does not support half-duplex connections).

Table 6-1. Data Communications Cables and Test Connectors

Cable No.	HP Part No.	Description
40242A	—	COMBINED RS232C and HP422 ADAPTER CABLE MALE 25 PIN TO FEMALE 50 PIN CABLE Provides for 50-pin cable connections Length: 1 meter (3 feet)
40242C	—	RS232C CABLE MALE 25 PIN TO FEMALE 25 PIN CABLE Length: 5 meters (16 feet)
40242M	—	US/EUROPEAN MODEM CABLE MALE 25 PIN TO MALE 25 PIN CABLE Male RS232C connector for interfacing the terminal to European telephone system via Bell 103 or 202C type European Modems. Length: 5 meters (16 feet)
40242P	—	HP DIRECT CONNECT TYPE 422 CABLE MALE 25 PIN TO MALE 5 PIN CABLE Length: 5 meters (16 feet)
40242X	—	HP DIRECT CONNECT TYPE 232 CABLE MALE 25 PIN TO MALE 3 PIN CABLE Used to interface the terminal to an HP 3000 Series 44 or 64. Length: 5 meters (16 feet)
40242Y	—	EMP PROTECT MALE 25 PIN TO MALE 25 PIN CABLE Male RS232C 25-pin connector for interfacing the terminal to an HP 1000, or 3000 Multiplexer. Provides protection from lightning-induced transients. For use in hardwired configurations only. Length: 5 meters (16 feet)
40242Z	—	RS232C MODEM BYPASS CABLE MALE 25 PIN TO FEMALE 25 PIN RS232C direct connect cable. Length: 5 meters (16 feet)
13222C	13222-60003	TERMINAL TO RS232 CABLE (used with 40242A cable) MALE 50 PIN TO FEMALE 25 PIN CABLE Female RS232C 25-pin connector. Length: 2 meters (6.6 feet)

Continued . . .

Table 6-1. Data Communications Cables and Test Connectors (Continued)

Cable No.	HP Part No.	Description
13222M	13222-60002	<p>EUROPEAN MODEM CABLE (used with 40242A cable) MALE 50 PIN TO MALE 25 PIN CABLE</p> <p>Male RS232C 25-pin connector for interfacing the terminal to the European telephone system via Bell 103 or 202C type European modems.</p> <p>Length: 5 meters (16 feet)</p>
13222N	13222-60001	<p>U.S. MODEM CABLE (used with 40242A cable) MALE 50 PIN TO MALE 25 PIN CABLE</p> <p>Male RS232C 25-pin connector for interfacing the terminal to an HP 1000, or 3000 Multiplexer; to a Bell 103A, 202C/D/S/T, 212A, or VADIC 3400 modem; or to an acoustic coupler (signal compatible only).</p> <p>Length: 5 meters (16 feet)</p>
13222P	—	<p>HP DIRECT CONNECT TYPE 422 CABLE (used with 40242A cable) MALE 50 PIN TO MALE 5 PIN CABLE</p> <p>Length: 5 meters (16 feet)</p>
13222W	13222-60007	<p>RS232C CABLE (used with 40242A cable) MALE 50 PIN TO FEMALE 25 PIN CABLE</p> <p>Female RS232C 25-pin connector for interfacing the terminal to an HP 3000 Computer System.</p> <p>Length: 5 meters (16 feet)</p>
13222X	—	<p>HP DIRECT CONNECT CABLE TYPE 232 CABLE (used with 40242A cable) MALE 50 PIN TO MALE 3 PIN CABLE</p> <p>Length: 5 meters (16 feet)</p>
13222Y	13222-60005	<p>EMP PROTECT (used with 40242A cable) MALE 50 PIN TO MALE 25 PIN CABLE</p> <p>Male RS232C 25-pin connector for interfacing the terminal to an HP 1000, or 3000 Multiplexer. Provides protection from lightning-induced transients. For use in hardwired configurations only.</p> <p>Length: 5 meters (16 feet)</p>
13222U	5061-2403	<p>MODEM BYPASS CABLE FEMALE 25 PIN TO FEMALE 25 PIN CABLE</p> <p>Crosses the signals so that two terminals can communicate with one another.</p> <p>Length: 1.5 meters (5 feet)</p>

Continued...

Table 6-1. Data Communications Cables and Test Connectors (Continued)

Cable No.	HP Part No.	Description
–	02620-60056	This is a male 50-pin test hood for use with 40242A cable.
–	02620-60062	This is a male RS232C test hood for use on port # 1.
–	02645-60004	This is a female RS232C test hood for use on an HP 13222M/N/Y or 40242M/Y datacomm cable.
–	5061-3248	This is a male HP422 test hood for use on port # 1.
–	5061-6703	This is a female (3 pin) RS232C test hood for use on an HP 13222X or 40242X datacomm cable.
–	5061-6704	This is a female (5 pin) HP422 test hood for use on an HP 13222P or 40242P datacomm cable.

## MODEM CONNECTIONS

If you decide on a point-to-point modem connection, you must now decide what type of modem to use. Note that point-to-point as supported by the terminal always employs asynchronous transmission. You will therefore be limiting your choice of modem to the asynchronous variety. Contact your local HP office for a list of supported modems.

### Modem Considerations

If you are communicating with the host computer through a modem, it may be necessary for you to turn on a modem power switch or make modem parity setting changes. The modem's baud rate and parity settings should be the same as those configured in the terminal.

Whenever the modem line (Data Set Ready) is active, and when the Asterisk field in the Datacomm Configuration Menu is set to "DM" (see Table 6-2), an asterisk appears below the space between the fourth and fifth screen labels in the Status Line at the bottom of the screen. If your facility requires the display of this "active modem" indicator, do not shut off the screen labels display.

The asterisk between the fourth and fifth screen label reflects the state of the Data Set Ready (DSR) input line to the terminal. When a modem is connected, the DSR line is high (active) and the modem indicator (asterisk) is on. When the DSR line is low, this signals a modem disconnect and the asterisk disappears from the screen.

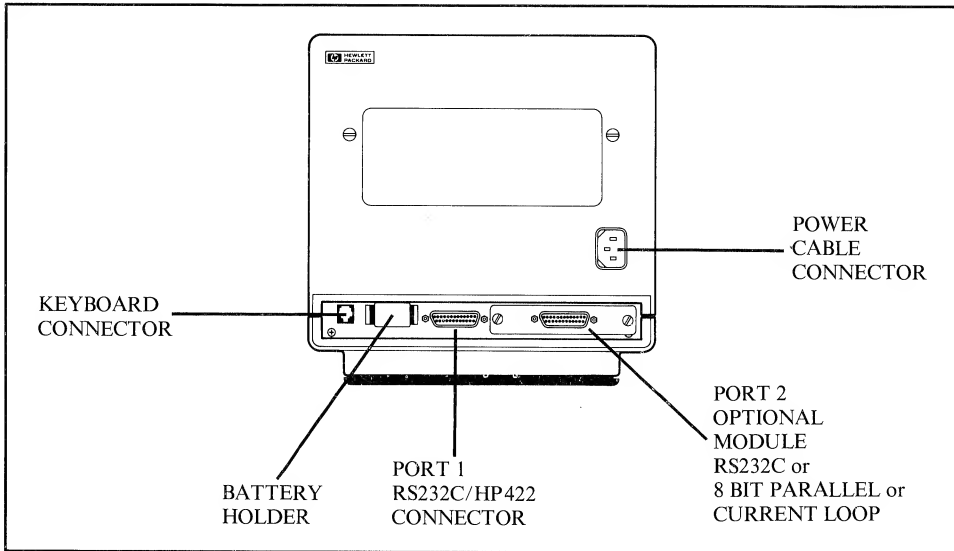


Figure 6-1. Terminal Rear View

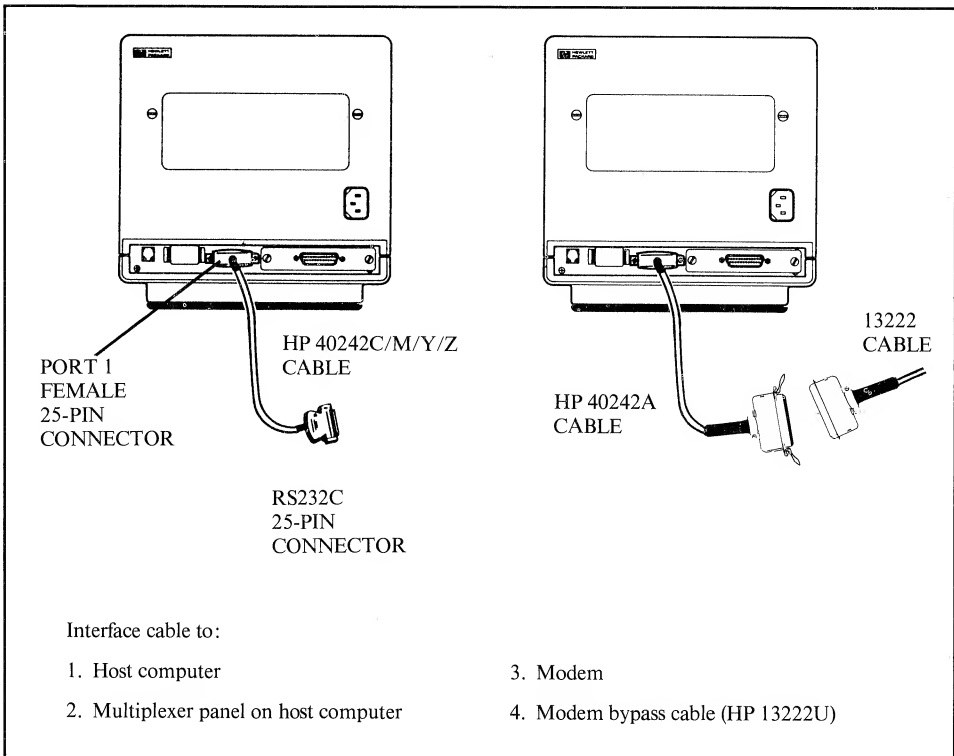


Figure 6-2. Terminal Cabling (HP 40242 and 13222 Cables)

## INSTALLING A POINT-TO-POINT CONFIGURATION

The terminal's datacomm port may be connected to a computer via a 25-pin, female RS232C/HP422 compatible connector provided on the back of the terminal (see Figure 6-1).

## CONFIGURING THE TERMINAL

Once the physical connections between the terminal and the computer or modem are complete, the terminal can be configured.

To configure the data communications portion of the terminal†, first use the following keystroke sequence:

[User System] , 

[f8]
config keys

 , 

[f3]
datacomm config

When you press the **datacomm config** ([f3]) function key, the datacomm configuration menu currently stored in non-volatile memory appears on the screen and the function key labels change to the following:

HP2392A:\*

[f1]	[f2]	[f3]	[f4]	[f5]	[f6]	[f7]	[f8]
SAVE CONFIG	NEXT CHOICE	PREVIOUS CHOICE	DEFAULT VALUES			DISPLAY FUNCTNS	config keys

Note that if you have not previously stored a menu in non-volatile memory, the default values are displayed in the configuration menu (see Figure 6-3).

The menu contains a set of unprotected fields that you access using the [Tab◀▶] key. You select the desired parameters in these fields using the **NEXT CHOICE** ([f2]) and **PREVIOUS CHOICE** ([f3]) function keys.

The meanings of the various fields in datacomm menu are described in Table 6-2.

When you have set all the fields to the desired values, you may then save them in non-volatile memory using the **SAVE CONFIG** ([f1]) function key. Note that when you do this, the particular datacomm configuration takes effect immediately.

### NOTE

Whenever a configuration menu is on the screen, incoming data, including escape sequences, is stored in buffer (up to 256 bytes) and not processed and no data is transmitted, until the menu is exited.

† To lock and unlock the datacomm configuration menu on the HP 2394A terminal, see Section II "Terminal Configuration Menu (HP 2394A)"

\*HP 2394A:

[f1]	[f2]	[f3]	[f4]	[f5]	[f6]	[f7]	[f8]
SAVE CONFIG	NEXT CHOICE	PREVIOUS CHOICE	DEFULAT VALUES	POWER ON VALUES	ACTIVE VALUES	DISPLAY FUNCTNS	config keys

While the datacomm configuration menu is displayed on the screen, the [f4] and [f8] keys have the effects described below:

[f4] DEFAULT VALUES	Pressing this key causes all fields in the menu to be filled with their default values.
[f8] config keys	Pressing this key removes the menu from the screen (WITHOUT activating it or saving it in non-volatile memory) and returns the function key labels to the following:

ext dev config label is blank when Port 2  
is not equipped with a datacomm module

[f1]	[f2]	[f3]	[f4]	[f5]	[f6]	[f7]	[f8]
		datacomm config	ext dev config	terminal config			

DATACOMM CONFIGURATION			
BaudRate	2400*	Parity/DataBits	None/8
Asterisk	OFF	Chk Parity	NO
RecvPace	None	SR (CH)	LO
XmitPace	None	EnqAck	YES
		CS (CB) Xmit	NO

Figure 6-3. Datacomm Configuration Menu

Table 6-2. Datacomm Configuration Menu Fields

<b>Baudrate</b>	<p>This field specifies at what speed you want the data transmission to take place.</p> <p>Values:    110            600            4800                     134.5        1200        9600                     150           1800        19200                     300           2400</p> <p style="text-align: center;">NOTE</p> <p>For 110 baud, the terminal is automatically configured to transmit 2 stop bits with the data to the computer. At 110 baud, the terminal also expects to receive 2 stop bits with the data received from the computer. For all other baud rates, 1 stop bit is transmitted with data and expected to be received with data.</p> <p>Operation at baud rates above 9600 may require handshaking or cpu-supplied delays.</p> <p>Default:    2400*</p>
-----------------	--

\* The default baud rate of the HP 2394A terminal is 9600.



Table 6-2. Datacomm Configuration Menu Fields (Continued)

<b>Parity/Data Bits</b>	<p>This field specifies what type of parity generation and checking you wish used with each data character and the number of bits per received or transmitted character. Note that when parity is set to NONE the number of Data Bits is always 8.</p> <p>Values: <b>None/8</b> (no parity bit)  <b>0's/7</b> (parity bit always zero)  <b>ODD/7</b> (odd parity)  <b>1's/7</b> (parity bit always one)  <b>EVEN/7</b> (even parity)</p> <p>For a complete discussion of "NONE/8" which chooses the Roman 8 character set, refer to Appendix B.</p> <p>Default: <b>None/8</b></p>
<b>EnqAck</b>	<p>This field enables or disables the use of the Hewlett-Packard ENQ-ACK handshake. This type of handshaking is described under "Pacing Mechanisms" in the "Point-to-Point Programming Information" portion of this section.</p> <p>Values: <b>YES</b> (enable) (default)  <b>NO</b> (disable)</p>
<b>Asterisk</b>	<p>This field specifies whether the transmit (modem) indicator in status line should be enabled or disabled and, if enabled, which RS232C control line it should reflect.</p> <p>The value "<b>OFF</b>" disables the TRANSMIT indicator altogether. The value "<b>CS</b>" specifies that the TRANSMIT indicator should reflect the state of the RS232C Clear to Send (CB) control line (asterisk = HI; no asterisk = LO). The value "<b>DM</b>" specifies that the TRANSMIT indicator should reflect the state of the Data Mode (DM) or RS232C Data Set Ready (CC) control line (asterisk = HI; no asterisk = LO).</p> <p>The value "<b>RR</b>" specifies that the transmit indicator should reflect the state of the receive ready (RR) or RS232C Data Carrier Detect (CF) control line (asterisk = HI; no asterisk = LO).</p> <p>Values: <b>OFF</b> (default) <b>CS</b> <b>DM</b> <b>RR</b></p>
<b>Chk Parity</b>	<p>This field is used for enabling or disabling the parity check feature for data characters received over the datacomm line. Note that if the <b>Parity</b> field (above) is set to <b>NONE</b>, then this field is ignored. Note that the specified parity bit is still transmitted.</p> <p>Values: <b>YES</b> (enable)  <b>NO</b> (disable) (default)</p>
<b>SR(CH)</b>	<p>This field specifies the desired state of the RS232C SR line when the terminal's power is first turned on or when the terminal is reset. The SR line, RS232C pin number 23, is defined as the Data Signal Rate Selector (DTE Source). It is normally used on dual speed modems to select the appropriate speed (single speed modems merely ignore this line).</p> <p>Values: <b>HI</b>  <b>LO</b> (default)</p>

Continued . . .

Table 6-2. Datacomm Configuration Menu Fields (Continued)

Recvpace	<p>Receive pacing is a mechanism by which the terminal automatically controls (halts and resumes) the transmission of data from the remote device. There is one way of performing receive pacing: by using the XON and XOFF control codes.</p> <p>If this field is set to <b>"XonXoff"</b>, the terminal will automatically perform receive pacing using XON (<math>\rho_1</math>) and XOFF (<math>\rho_3</math>) control codes. With this type of receive pacing, the terminal causes the remote device to halt transmission by sending an XOFF code and to resume transmission by sending an XON code. For this type of receive pacing to work, the remote device must of course be configured to start and stop transmission in response to XON and XOFF codes.</p> <p>Note that if the remote device recognizes XON and XOFF codes and your terminal is operating in character mode, you can issue the codes through the keyboard regardless of the setting of this field. The [CTRL] and Q keys (when pressed simultaneously) generate an XON code and the [CTRL] and S keys generate XOFF when XON/XOFF receive pacing is on. The [STOP] key can also be used to stop the flow of data from the input buffer to the screen and then resume it.</p> <p>Values:   <b>NONE</b> (default)               <b>XON/XOFF</b></p>
CS(CB)Xmit	<p>This field specifies whether or not an "on" state (+ 12 V) on the RS232C Clear to Send (CS/CB) control line is a required condition for transmitting data. For a modem configuration, it is recommended that you set this field to <b>"YES"</b>.</p> <p>Values:   <b>YES</b>               <b>NO</b> (default)</p>
XmitPace	<p>Transmit pacing is a mechanism by which the remote device can control (stop and resume) the transmission of data from the terminal.</p> <p>If enabled, transmit pacing is performed using XON and XOFF control codes. When the terminal receives an XOFF code (<math>\rho_3</math>), it stops transmitting data. When the terminal subsequently receives an XON code (<math>\rho_1</math>), it resumes transmitting data. This type of handshaking can be used in conjunction with D1/D2 handshaking.</p> <p>If this field is set to <b>"NONE"</b>, the terminal does NOT recognize the <math>\rho_1</math> and <math>\rho_3</math> codes as XON and XOFF.</p> <p>For another form of transmit pacing, refer to the description of the CS(CB)Xmit field above.</p> <p>Values:   <b>NONE</b> (default)               <b>XON/XOFF</b></p>

## POINT-TO-POINT PROGRAMMING INFORMATION

This topic discusses programming information of interest to someone who is writing a data communications driver or controller program to communicate with the terminal in an asynchronous point-to-point environment.

An asynchronous point-to-point data communications environment is characterized by a flow of characters that have been produced over random time intervals. In order to achieve hardware synchronization, each character is delimited by a “start bit” and one or more “stop bits”.

### Start And Stop Bits

These hardware-generated bits are used for synchronizing the transmit and receive devices in an asynchronous environment. A start bit is a space or “zero” line state (+ 12 V) that lasts for 1.0 bit time; it is affixed to the beginning of a serial character bit stream (which may also include a parity bit). A stop bit is a mark or a “one” line state (– 12 V) that lasts for 1.0 bit time; it is appended to the end of each serial character by stream. After the stop bit, the line remains in the mark state until the next character, signified by a start bit, is transmitted.

The start and stop bits are not configurable. For 110 baud, the terminal is automatically configured to transmit 2 stop bits with the data to the computer. At 110 baud, the terminal also expects to receive 2 stop bits with the data received from the computer. For all other baud rates, 1 stop bit is transmitted with data and 1 stop bit is expected to be received with data.

### Parity Checking

In an asynchronous point-to-point environment, the terminal provides a vertical redundancy check (VRC), which is a character-based error checking mechanism for non-binary data. With VRC, an additional bit is affixed to each character to provide an expected high-order bit state for each character. This type of parity generation and checking is a means of determining the validity of data transfer on a character-by-character basis.

Note that when 8-bit data is being exchanged, parity cannot be used and is automatically set to “NONE” in the datacomm configuration menu.

The terminal offers the following five types of parity:

1. 0's.        The high-order bit is always a zero.
2. 1's.        The high-order bit is always a one.
3. ODD.       The high-order bit is set to a zero or a one, whichever produces an odd number of one bits in the overall character representation (the seven data bits plus the eighth parity bit).
4. EVEN.      The high-order bit is set to a zero or a one, whichever produces an even number of one bits in the overall character representation (the seven data bits plus the eighth parity bit).
5. NONE.      Eight bits of data are transmitted and received. No parity bit is transmitted or received.

See Appendix B for ROMAN 8 transmissions.

## Receive Buffer

The terminal's receive buffer is a first in/first out (FIFO) storage area for accepting data from the remote device. When you are using any type of receive pacing, this buffer is partitioned into a working buffer and a 40-byte overrun area. For example, the specified buffer size is always 256 bytes, thus if receive pacing is being used, the working buffer is 216 bytes long and the overrun area is 40 bytes long. When the data being received exceeds the working buffer and intrudes on the overrun area, the terminal will exercise its receive pacing mechanism (send an XOFF, for example, if XON/XOFF receive pacing is enabled) at that time to temporarily halt the flow of data from the remote device. When enough data has been processed so that the receive buffer is only quarter full (60 characters), the terminal then signals the remote device to resume transmission (by sending an XON, for example, if XON/XOFF receive pacing is enabled).

There is no equivalent overrun area for transmitting data from the terminal to the remote device.

## Receive Errors

When receiving data from the remote device, the terminal can detect the following three types of error conditions (in addition to parity errors):

1. Character overruns – a character is received before the preceding character was processed by the terminal's datacomm firmware.
2. Framing errors – no stop bit was detected at the end of a character.
3. Buffer overflows – the entire allocated buffer space is filled (both the working buffer and the overrun area).

Receive errors, when detected, are reported to the remote device by way of byte 5 of the primary terminal status bytes. The remote device will not be able to determine which type of error occurred. If multiple receive errors occur simultaneously, only one will be reported.

When a datacomm receive error occurs, a delete character (■) is placed in the datacomm queue and later it is displayed on the destination device.

#### NOTE

Because NULL and DEL characters are automatically stripped from datacomm, the only DEL characters appearing on the screen from datacomm are the result of datacomm errors.

### Local/Remote Mode

The data communications portion of the terminal operates independently whether the terminal is in local or remote mode. If the terminal is switched from remote to local while data is being received from the remote device, the datacomm portion of the terminal continues receiving data (it does NOT halt the transmission). In such a case, the data received while the terminal is in local is discarded by the terminal's maincode firmware.

### Full-duplex Operation

In a full-duplex environment, the terminal is capable of transmitting and receiving data simultaneously. The ability to transmit may be inhibited temporarily, but it is never exclusive of the ability to receive. Two physical sets of data lines are required; control lines are needed only when hardware handshaking or a modem is used. Transitions on the control lines have no effect on the actual transmit/receive state of the terminal.

When the terminal is connected to the host computer via a modem, the following primary control lines are required:

- Request to Send (RS/CA)

- Clear to Send (CS/CB)

- Data Terminal Ready (CD/TR)

If the terminal is hardwired directly to an HP 3000 computer system (no modem), only Transmit Data (SD/BA), Receive Data (RD/BB), and Signal Ground (AB) are required.

## Pacing Mechanisms

In a full-duplex environment, the terminal can participate in either of the following forms of transmit pacing:

1. **Hardware handshake.** The host computer can temporarily restrain the terminal from transmitting by lowering the Clear to Send (CB) line. Note that this type of transmit pacing can only be used in a hardwired configuration where the Clear to Send (CB) line exists in the cabling.
2. **XON/XOFF handshake.** The host computer or external printer uses the control codes **XON** (D<sub>1</sub>) and **XOFF** (D<sub>3</sub>) to start and stop the terminal from transmitting. Note that a single **XON** code cancels any number of preceding **XOFF** codes.


In a full-duplex environment, the terminal can also participate in the XON/XOFF handshake form of receive pacing, in which the terminal uses the control codes **XON** (D<sub>1</sub>) and **XOFF** (D<sub>3</sub>) to start and stop the host computer from transmitting. Note that a single **XON** code cancels any number of **XOFF** codes.

The terminal can also participate in an ENQ/ACK handshake (which is a Hewlett-Packard handshaking mechanism). With this form of handshaking, the host computer transmits a block of data and then sends an ASCII <ENQ> control code. The terminal responds to the <ENQ> by sending back an ASCII <ACK> control code when it has processed all of the data preceding the <ENQ>. The general interpretation of these two control code is as follows:

ENQ: "Have you processed the data up to this point?"

ACK: "Yes, I have."

The above pacing mechanisms are responded to by the terminal in the following order of precedence:

1. Hardware handshaking pacing (highest priority)
  2. XON/XOFF transmit pacing
  3. XON/XOFF receive pacing
  4. ENQ/ACK pacing (lowest priority)
- 

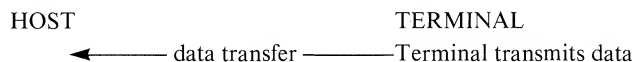
## Transfers And Handshaking

The terminal provides three types of data transfer operations:

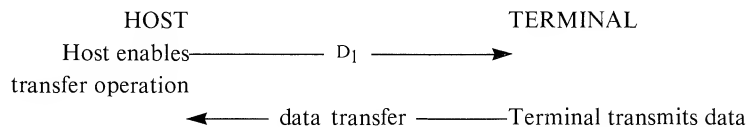
1. Short block transfer. Used in data transfer operations involving:
  - a) Terminal primary status (response to  $E_C^{\wedge}$ ).
  - b) Terminal secondary status (response to  $E_C^{\sim}$ ).
  - c) Device status (response to  $E_C \& p4^{\wedge}$ ).
  - d) Cursor sense, relative or absolute (response to  $E_C^{\sim}$  or  $E_C a$ ).
  - e) Transmit only user key (T) or [Select] key when the terminal is in block line mode or character mode.
  - f)  $E_C d$  (initiate data transfer).
  - g) Device completion status (S, F or U).
  - h) Terminal ID status (terminal response to  $E_C * s^{\wedge}$ ).
2. Long block transfer. Used in data transfer operations involving:
  - a) Transmit only user key (T) or [Select] key when the terminal is in block page mode.
  - b) [Enter] key in block mode (line or page).
3. Long character transfer. Used in data transfer operations involving:
  - a) [Enter] key in character mode.
  - b) [Return] or [Enter] key in modify mode.

In addition, the terminal provides three types of handshaking:

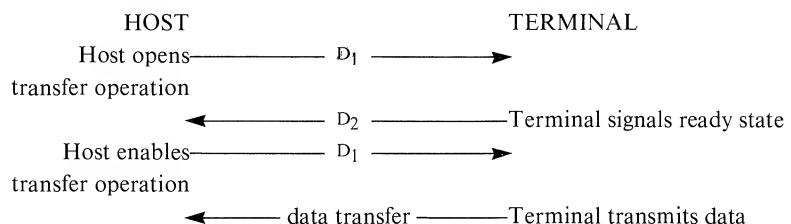
Type 1 – No handshake



Type 2 –  $D_1$  handshake



Type 3 -  $D_1/D_2/D_1$  handshake



The handshake used depends on the type of data transfer operation, and the setting of the “InhHndShk (G)” and “InhDC2 (H)” fields in the terminal configuration menu (see Section II). Table 6-3 lists the available handshakes.

Table 6-3. DC1/DC2 Handshake Used In Data Transfer

InhHndShk (G)	InhDC2 (H)	Short Block Transfer	Long Block Transfer	Long Character Transfer
		HANDSHAKE		
NO (0)	NO (0)	Type 2 (DC1)	Type 3 (DC1/DC2/DC1)	Type 1 (no handshake)
NO (0)	YES (1)	Type 2 (DC1)	Type 1 (no handshake)	Type 1 (no handshake)
YES (1)	NO (0)	Type 3 (DC1/DC2/DC1)	Type 3 (DC1/DC2/DC1)	Type 3 (DC1/DC2/DC1)
YES (1)	YES (1)	Type 1 (no handshake)	Type 1 (no handshake)	Type 1 (no handshake)

## Data Transfer Priority

If more than one transfer request is pending at the same time, the execution priority is:

1. Primary status.
2. Secondary status.
3. Device status.
4. Cursor sense.
5. Transmit user key (T) or [Select] key.
6. [Enter] or Fcd.
7. Device completion status (S, F or U).
8. ID status.



## INTRODUCTION

This section tells how a program executing in a host computer obtains and interprets status information from the terminal.

Status requests are issued in the form of escape sequences. There are four types of status requests:

1. Terminal ID Status. This request is the means by which your program verifies what kind of terminal it is communicating with.
2. Primary Terminal Status. This request returns seven bytes that report the status of some of the latching keys, various error and pending flags, and the following configuration menu fields (see Section II, table 2-1 for descriptions of the fields):

<b>XmitFnctn (A)</b>	<b>SPOW (B)</b>	<b>InhEolWrp (C)</b>	<b>Line/Page (D)</b>
<b>InhHndShk (G)</b>	<b>InhDC2 (H)</b>	<b>EscXfer (N)</b>	

3. Secondary Terminal Status. This request returns seven bytes that report the status of the memory lock, buffer memory, and I/O firmware.
4. Device Status. This request returns three bytes that report the status of the external printer.

The escape sequence used for each of the above requests and the format of the returned status information is presented in the following paragraphs.

All status requests are treated as block transfers. In response to a status request, the terminal transmits an escape sequence, followed by a series of bytes, followed by a terminator. The terminator is as follows:

Character Mode: **C<sub>R</sub>** or **C<sub>R</sub> L<sub>F</sub>**  
 Block Line Mode: **C<sub>R</sub>** or **C<sub>R</sub> L<sub>F</sub>**  
 Block Page Mode: **< Blkterminator >**

In either character mode or block line mode, the **C<sub>R</sub> L<sub>F</sub>** is used if auto line feed mode is enabled. In block page mode, the block terminator is **< Blk Terminator > (R<sub>S</sub>)**.

The type of handshaking used is determined by the setting of the **InhHndShk** and **Inh DC2** fields of the configuration menu as follows:

1. **InhHndShk (G)** = YES                      }  
    **Inh DC2 (H)**    = YES                      } No handshake
2. **InhHndShk (G)** = NO                     }  
    **Inh DC2 (H)**    = YES or NO            } D<sub>1</sub> handshake
3. **InhHndShk (G)** = YES                    }  
    **Inh DC2 (H)**    = NO                    } D<sub>1</sub>/D<sub>2</sub>/D<sub>1</sub> handshake

## INTERPRETING STATUS

For primary, secondary, and device status requests, the terminal returns an escape sequence followed by a string of bytes. The status information is contained in the lower four bits of each byte. The upper four bits are set so that the byte translates into one of the 16 characters shown in table 7-1.

For a terminal ID request, the terminal returns the 5-character defined in the terminal configuration menu.

### Terminal ID Status

You request the terminal ID status by issuing the following escape sequence:

**ESC\*s^**

The terminal responds by sending back the following five-character string defined in the **Terminal Id** field in the terminal configuration menu (see Section II). The default value is:

**2392A if you have an HP 2392A terminal**

**2394A if you have an HP 2394A terminal**

Table 7-1. Status Characters

CHARACTER	BINARY EQUIVALENT
0	0011 0000
1	0011 0001
2	0011 0010
3	0011 0011
4	0011 0100
5	0011 0101
6	0011 0110
7	0011 0111
8	0011 1000
9	0011 1001
:	0011 1010
;	0011 1011
<	0011 1100
=	0011 1101
>	0011 1110
?	0011 1111

## Terminal Status

Terminal status is made up of 14 status bytes (bytes 0-13) containing information such as display memory size, switch settings, configuration menu settings, and terminal errors. These 14 status bytes are displayed below the self-test screen pattern when the “**TERMINAL TEST**” ([f5]) key (in the “service keys” set of function keys) is pressed. There are two terminal status requests: primary and secondary. Each returns a set of 7 status bytes.

**PRIMARY TERMINAL STATUS.** You request the first set of terminal status bytes (bytes 0-6) by issuing the following escape sequence:

$E_C \wedge$

The terminal responds with an  $E_C \backslash$ , and seven status bytes followed by a terminator. A typical primary terminal status request and response is illustrated in figure 7-1. The example assumes that the  $D_1$  handshake is being used and that the appropriate terminator is a  $Q_R$ .

**SECONDARY TERMINAL STATUS.** You request the second set of terminal status bytes (bytes 7-13) by issuing the following escape sequence:

$E_C \sim$

The terminal responds with an  $E_C |$ , and seven status bytes followed by a terminator. A typical secondary terminal status request and response is illustrated in figure 7-2. The example assumes that the  $D_1$  handshake is being used and that the appropriate terminator is a  $Q_R$ .

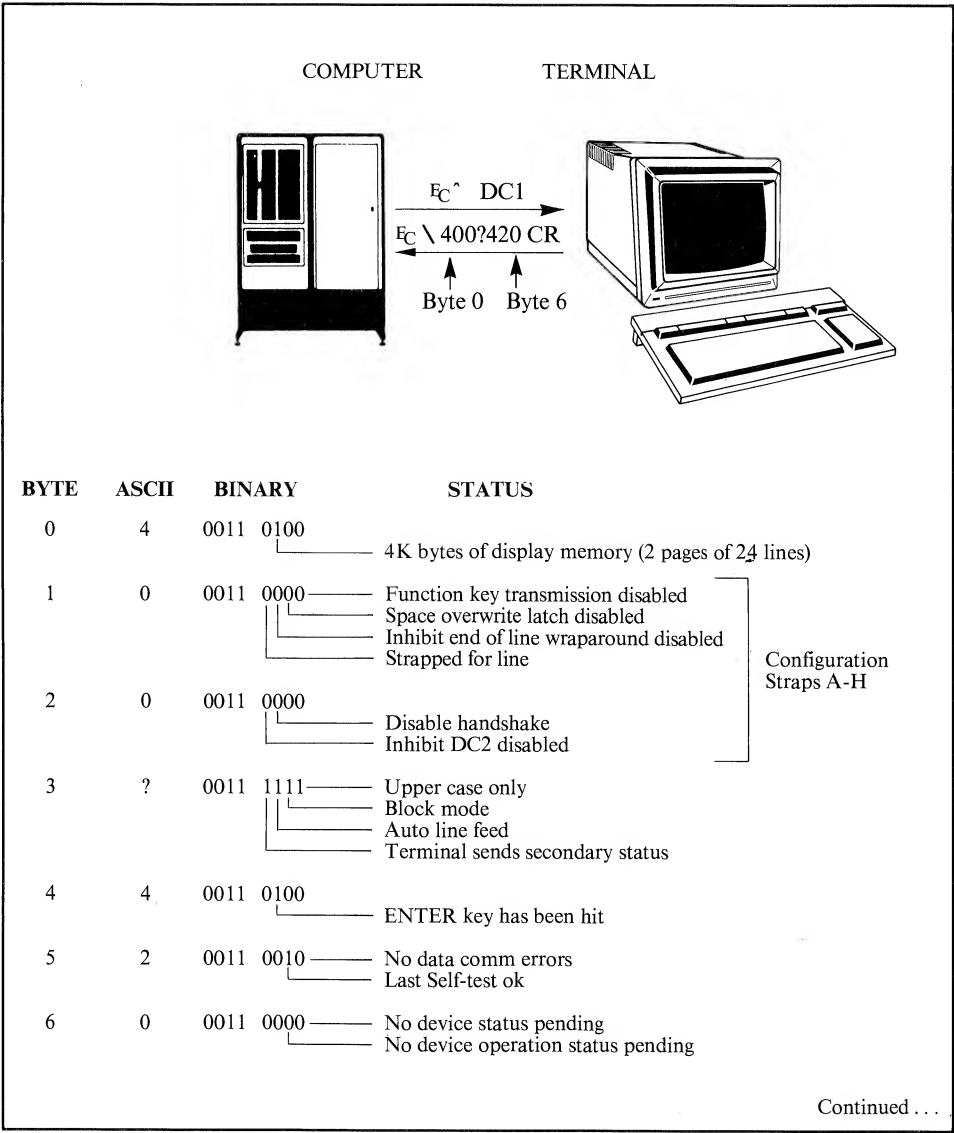
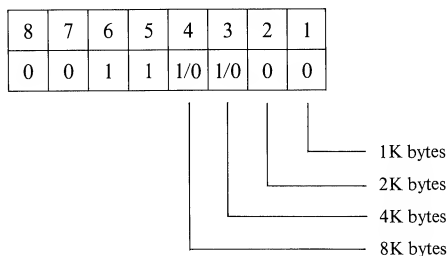
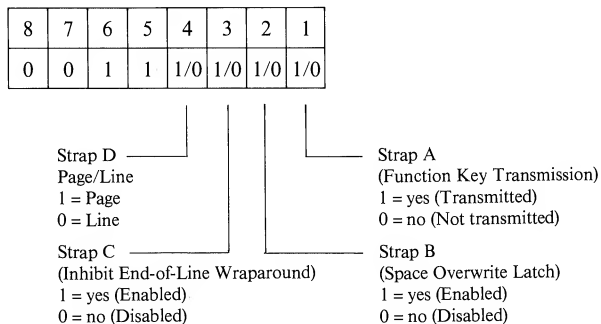


Figure 7-1. Primary Terminal Status Example

**PRIMARY STATUS BYTES****BYTE 0      DISPLAY MEMORY SIZE**

This byte specifies the amount of display memory available in the terminal.

Note that "1111" specifies 15K or more bytes.

**BYTE 1      CONFIGURATION STRAPS A-D**

Refer to Section II for a detailed description of configuration straps A-D

Continued . . .

Figure 7-1. Primary Terminal Status Example (Continued)

**BYTE 2      CONFIGURATION STRAPS E-H**

8	7	6	5	4	3	2	1
0	0	1	1	1/0	1/0	0	0

Strap H (Inhibit DC2) ——— Strap E  
 1 = yes (Enabled)  
 0 = no (Disabled)

Strap G (DC2 Handshake) ——— Strap F  
 1 = yes (Enabled)  
 0 = no (Disabled)

Refer to Section II for a detailed description of configuration straps G and H.  
 Straps E and F do not apply to the terminal.

**BYTE 3      LATCHING KEYS**

8	7	6	5	4	3	2	1
0	0	1	1	1	1/0	1/0	1/0

Terminal sends secondary status ——— CAPS LOCK field in terminal configuration menu  
 1 upper case only  
 0 upper and lower case

AUTO LF Key ——— BLOCK MODE Key  
 1 = auto LF  
 0 = no auto LF

1 block mode  
 0 character mode

**BYTE 4      TRANSFER PENDING FLAGS**

8	7	6	5	4	3	2	1
0	0	1	1	1/0	1/0	1/0	1/0

Secondary Status Pending ——— Cursor Sense pending  
 1 yes  
 0 no

ENTER Key Pending ——— Function Key Pending  
 1 yes  
 0 no

Continued ...

Figure 7-1. Primary Terminal Status Example (Continued)

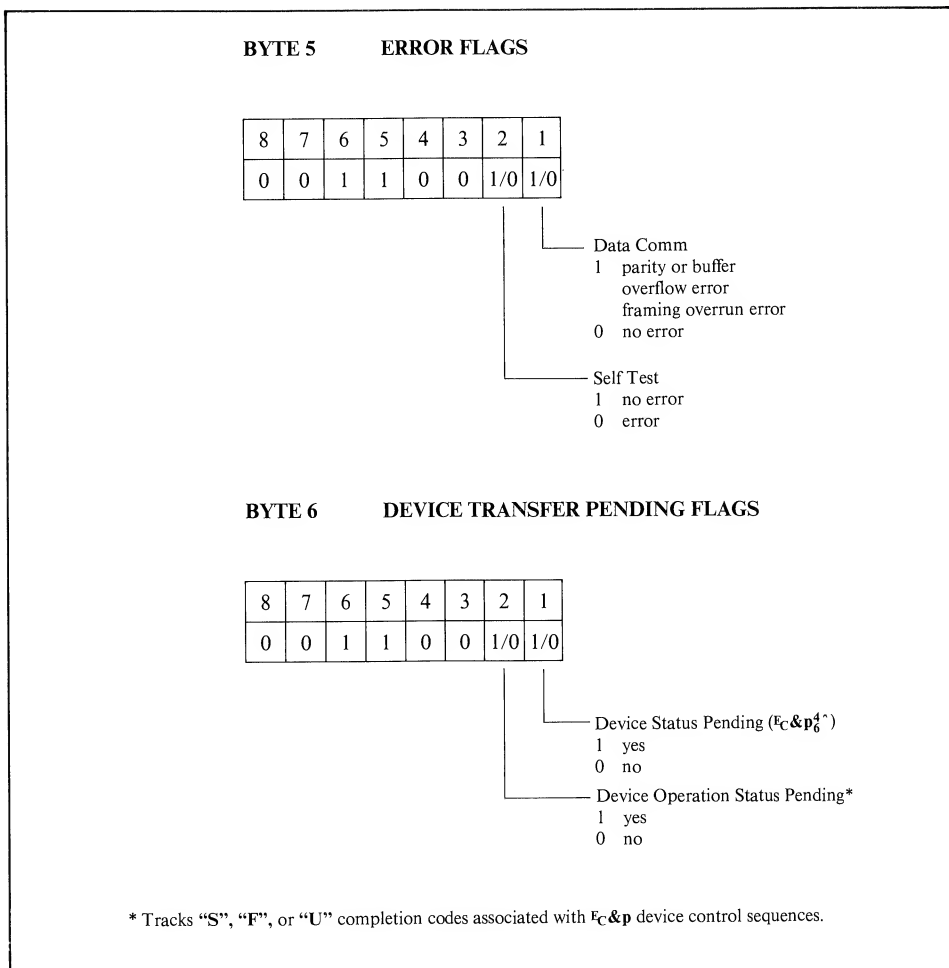


Figure 7-1. Primary Terminal Status Example (Continued)

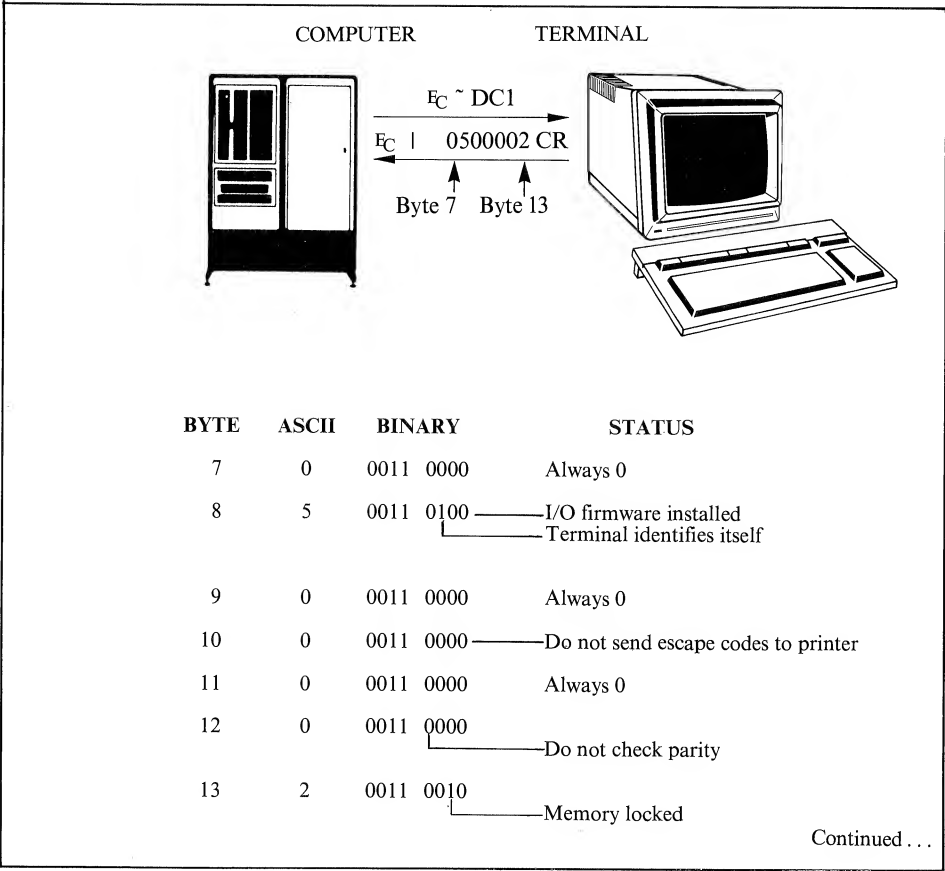


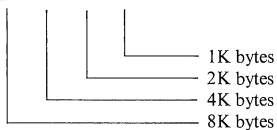
Figure 7-2. Secondary Terminal Status Example



## SECONDARY STATUS BYTES

### BYTE 7      BUFFER MEMORY (always zero)

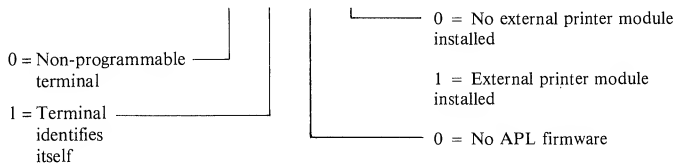
8	7	6	5	4	3	2	1
0	0	1	1	0	0	0	0



Memory installed in addition to display memory that is available for use as data buffers. Note that the HP 2392A terminal always return a 0 value.

### BYTE 8      TERMINAL FIRMWARE CONFIGURATION

8	7	6	5	4	3	2	1
0	0	1	1	0	1	0	0

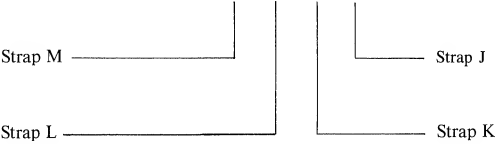


Continued...

Figure 7-2. Secondary Terminal Status Example (Continued)

**BYTE 9      CONFIGURATION STRAPS J-M**  
(always zero)

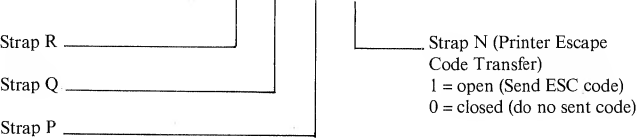
8	7	6	5	4	3	2	1
0	0	1	1	0	0	0	0



Straps J-M do not apply to the terminal.

**BYTE 10      KEYBOARD INTERFACE KEYS (N-R)**

8	7	6	5	4	3	2	1
0	0	1	1	0	0	0	1/0



Straps P-R do not apply to the terminal.

**BYTE 11      CONFIGURATION STRAPS S-V**  
(always zero)

8	7	6	5	4	3	2	1
0	0	1	1	0	0	0	0



Straps S-V do not apply to the terminal.

Continued ...

Figure 7-2. Secondary Terminal Status Example (Continued)

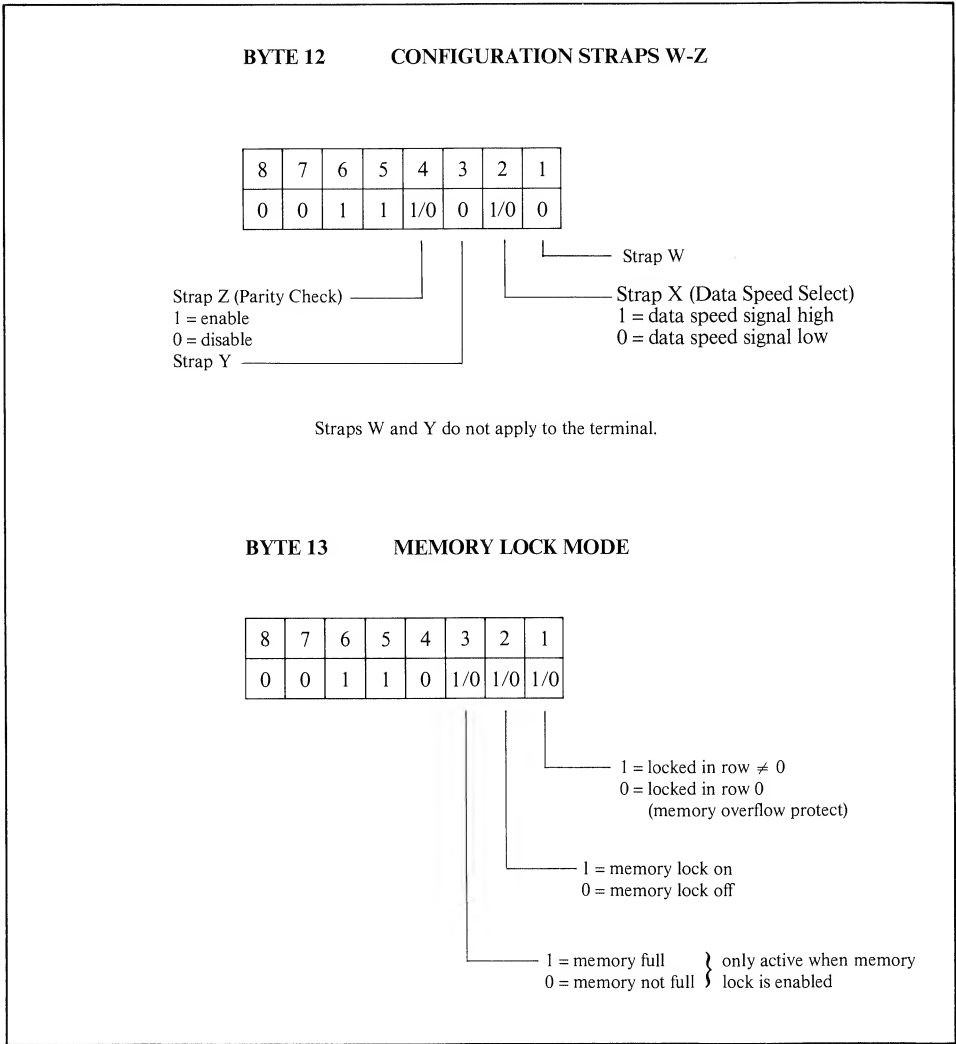


Figure 7-2. Secondary Terminal Status Example (Continued)

## TERMINAL CAPABILITIES (HP 2394A)

Four requests can be issued for terminal capabilities: alphanumeric, graphics, interface capabilities, and amount of RAM memory. These requests are generated with the following escape sequence:

$E_c *s <x>^{\wedge}$

where “x” selects the request type, as follows:

### X. Requested Information

- 1 Alphanumeric capabilities.
- 2 Graphics capabilities.
- 3 Amount of RAM memory.
- 4 Interface capabilities.

The terminal responds with a string of bytes. The first byte indicates the number of status bytes in the response (this byte does not include itself in the count). The following byte(s) contain the requested data (Figures 7-3 through 7-7).

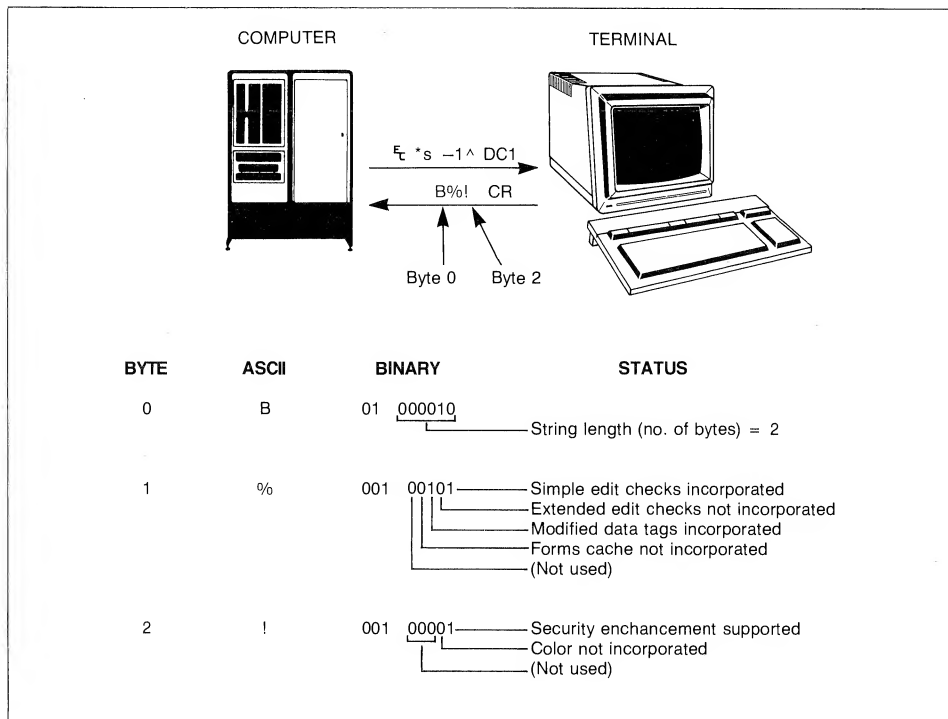


Figure 7-3. Terminal Capabilities (Alphanumeric-Typical) Status Example

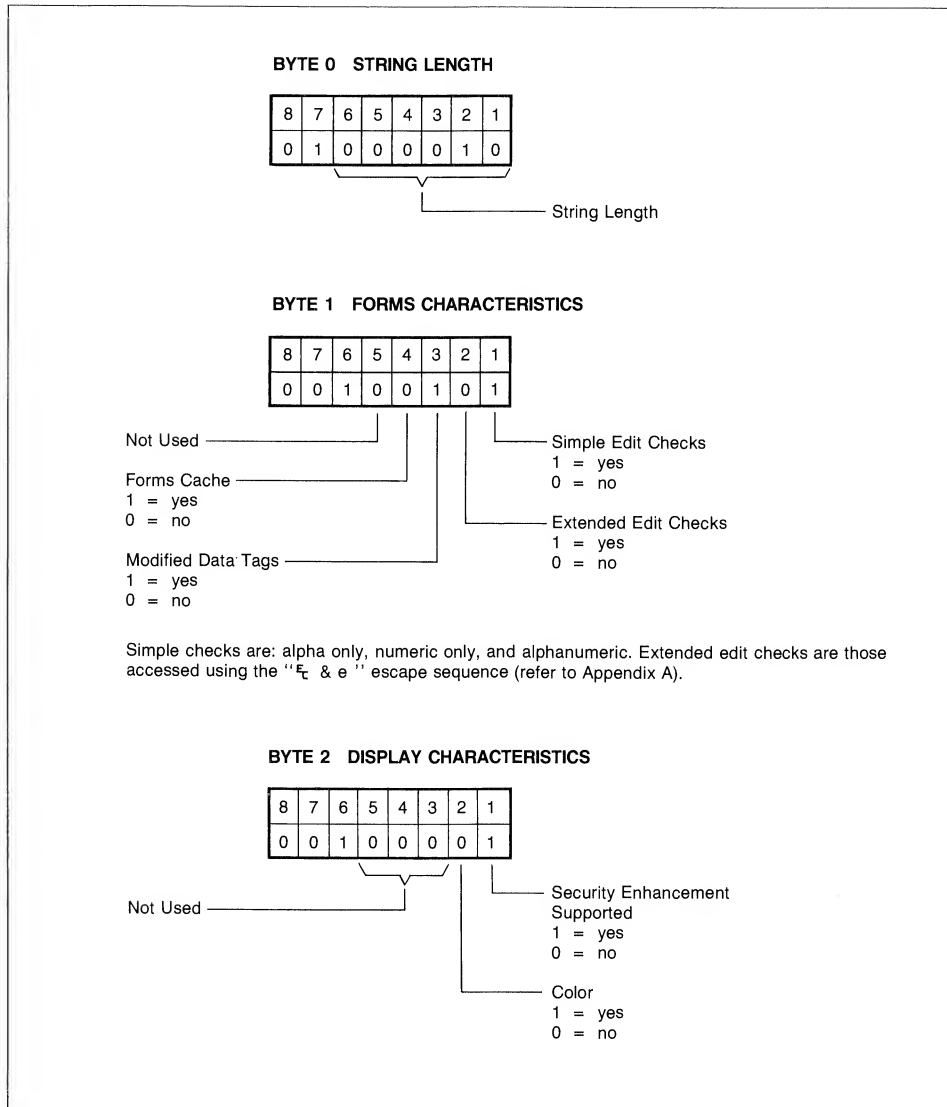


Figure 7-4. Terminal Alphanumeric Capabilities Status Bytes

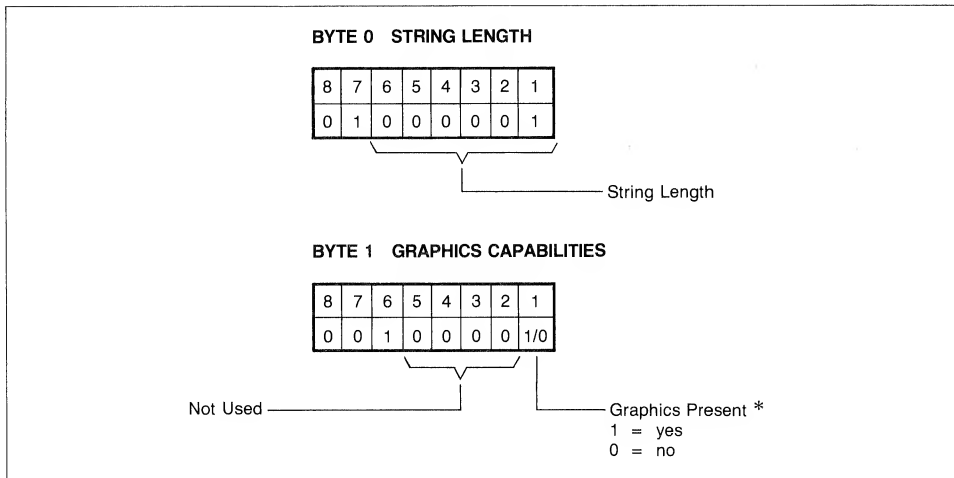


Figure 7-5. Terminal Graphics Capabilities Status Bytes

\* The HP 2392A/94A terminals do not have graphics capabilities.

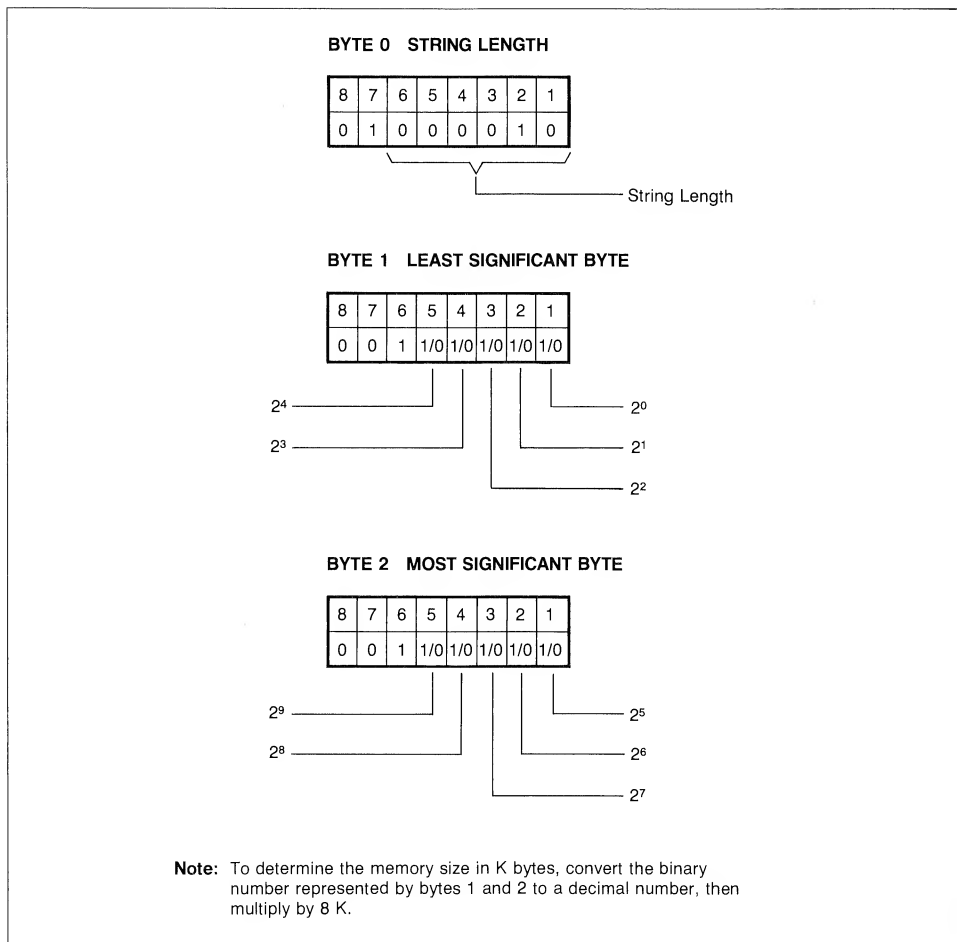


Figure 7-6. Installed Memory Status Bytes

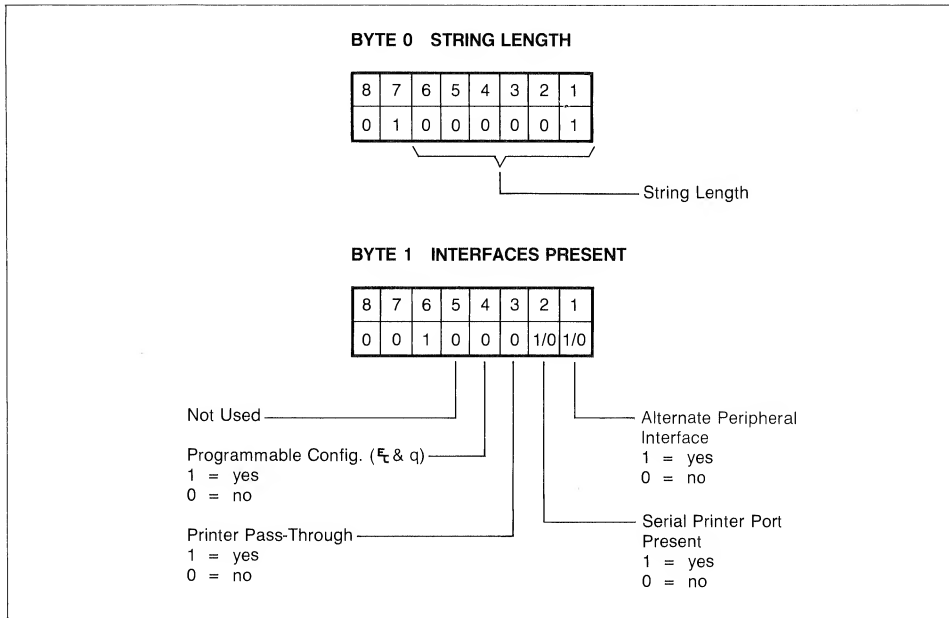


Figure 7-7. Terminal Interface Capabilities Status Bytes



## DEVICE STATUS

The status of the external printer can be obtained by issuing a device status request. This request would typically be made following a print operation or after examining byte 6 of the primary status. The device status bytes are shown in Figure 7-8.

You request device status by issuing the following escape sequence:

$\text{E}_C \& \text{p} \text{ } \langle \text{device code} \rangle \text{ } ^$

where  $\langle \text{device code} \rangle$  is either 4 or 6. The external printer is 4 on the HP 2392A. If  $\langle \text{device code} \rangle = 6$  (internal printer), the terminal will respond by indicating there is no internal printer.

If  $\langle \text{device code} \rangle$  is any value other than 4 or 6, the escape sequence is ignored.

The terminal responds with the sequence  $\text{E}_C \text{ p } \langle \text{device code} \rangle$ , followed by three status bytes followed by a terminator (where  $\langle \text{device code} \rangle = 4$ ).

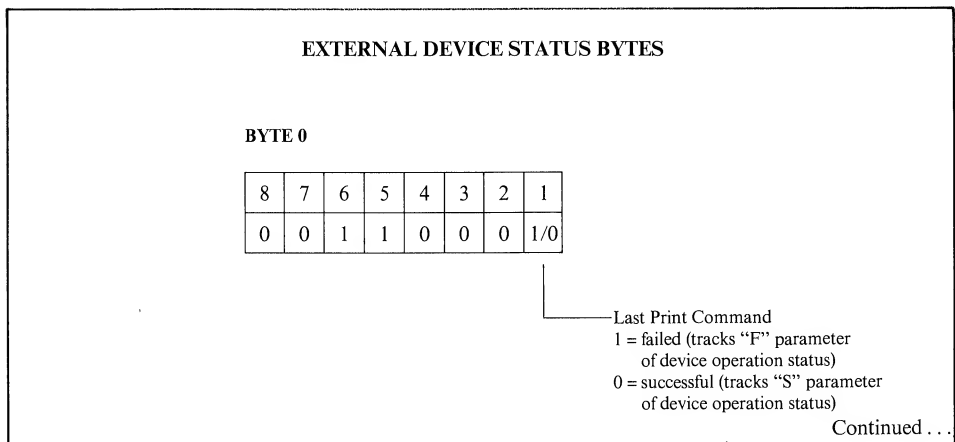


Figure 7-8. External Device Status Bytes

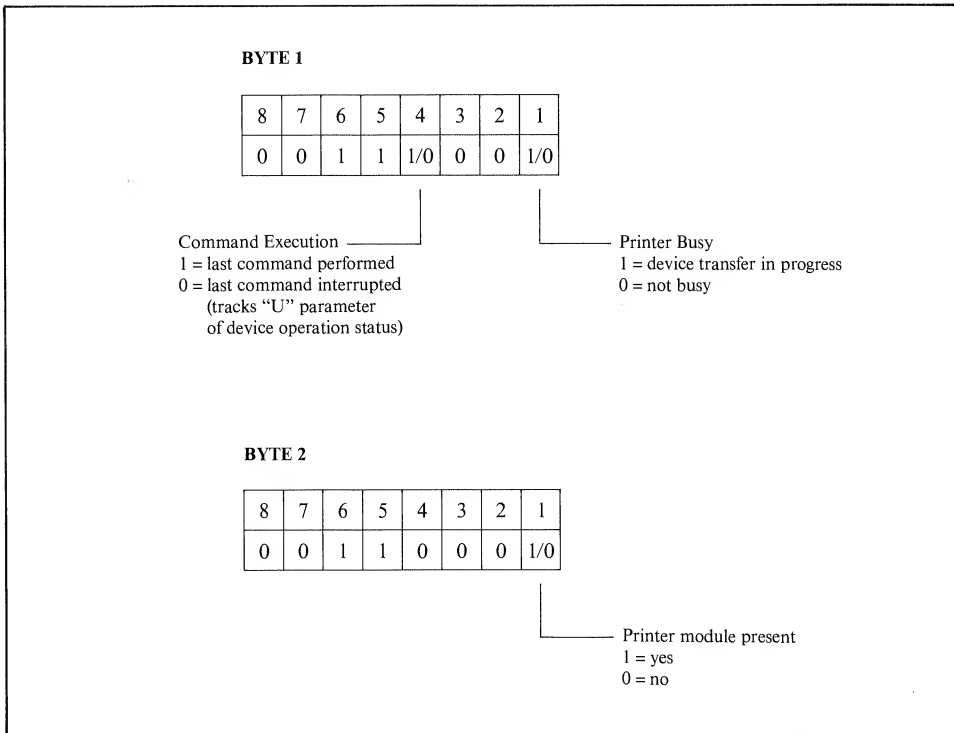


Figure 7-8. External Device Status Bytes (Continued)

## FORMS CACHE STATUS

A device status request may be used to determine the amount of forms cache currently allocated, how much cache is available for storing additional forms, and whether or not a specific form is currently defined. There are three types of forms cache status escape sequences that can be used:

$E_C \& p9 \wedge$

or

$E_C \& p < \text{form\#} > p9 \wedge$

or

$E_C \& p < \text{form name} > n9 \wedge$

With each sequence, the terminal will respond with a message containing forms cache status information. The format of the response is shown below:

$E_C \backslash p9 < \text{status byte 0} > < \text{status byte 1} > < \text{status byte 2} > < \text{terminator} >$

Each status byte contains status information in the lower 4 bits. The upper 4 bits are set to 0011. This insures that the character code for each byte is always in the range of 30H to 3FH (0-9;; < = >?).

Each status byte contains status information in the lower 4 bits. The upper 4 bits are set to 0011. This insures that the character code for each byte is always in the range of 30H to 3FH (0-9;; < = >?).

Status byte 0 and 1 contain block count information. Status byte 2 defines the state (present/not present) of a specific form. When a form number or name is included in the status request, the block count is the amount of space remaining to store additional forms within the allocated forms cache memory. Status byte 2 indicates if the form is present (1) or undefined (0).

If a form number is not included in the status request, then status bytes 0 and 1 indicate the number of blocks currently allocated to forms cache. In this case, status byte 2 is always set to 0.

The following example assumes that 10 blocks have been allocated for forms cache. In addition, form 50 has been successfully downloaded to cache memory and occupies 3 blocks. The form name of this form has been defined as RECEIPTS. The current terminator is assumed to be a carriage return ( $C_R$ ). For each status request, the response provided by the terminal is shown:

$E_C \& p9 \wedge$	request status of cache
$E_C \backslash p90:0C_R$	terminal response (0: = 10 blocks allocated)
$E_C \& p50p9 \wedge$	request status of form 50
$E_C \backslash p9071C_R$	terminal response (071 = 7 blocks available, form 50 present)
$E_C \& p < \text{RECEIPTS} > n9 \wedge$	request status of form RECEIPTS

$E_C \backslash p9071 C_R$ 

terminal response

(071 = 7 blocks available, form RECEIPTS present)

The terminal response is returned using the handshaking currently defined for the terminal.

## READING THE FORMS CACHE DIRECTORY

The forms cache directory allows an application program to determine exactly which forms are currently stored in forms cache. This provides an application program with the ability to easily determine if any or all of its associated forms have already been downloaded to cache memory. The feature is useful when several different programs in an application share one or more common forms, or an application is stopped and restarted on the same terminal.

To read the forms cache directory, an extension of the device status request (escape sequence) is used. The sequence is shown below:

 $E_C \& p < > n9 \wedge$ 

This status request will cause the terminal to return a list of the forms currently stored in cache memory. The list will contain form numbers and names (if defined). The format of the list is as follows:

 $E_C \backslash p9 < fnum_1 > < < fname_1 > > \dots < terminator >$ 

The form number ( $< fnum_n >$ ) and form name ( $< < fname_n > >$ ) of each form are returned. If a form has not been assigned a name, the “<” and “>” characters will appear in the list to indicate a null form name. Form numbers and names are returned in the same sequence that they were defined. For example, if form 3 was defined with the name of ORDERS, and form 1 was subsequently defined with no name, the status request and response would appear as follows:

 $E_C \& p < > n9 \wedge$ 

status request from host

 $E_C \backslash p93 < ORDERS > 1 < > C_R$ 

terminal response

form 3 is defined with name ORDERS

form 1 is defined

# Error Messages and Self-Test

SECTION

VIII

## INTRODUCTION

This section is divided into two portions. The first discusses the various error messages that may appear on the terminal's screen while you are attempting to perform operations through the keyboard. The second discusses the various types of self-tests that are incorporated into the terminal.

## ERROR MESSAGES

When the terminal detects a parameter inconsistency or error condition, it locks the keyboard and displays an appropriate error message across the bottom of the screen (replacing the function key labels). Pressing **[Return]** will unlock the keyboard, clear the message, and reinstate the current function key labels.

The various possible error messages and their general meanings are as follow:

### **Default configs used**

**Press RETURN to clear**

This message is displayed when the terminal attempts to read the content of non-volatile memory but detects a CRC error (e.g., at power-on time, during a hard reset).

The “**default config used**” power-on message may indicate that the batteries need to be changed. Instructions on how to change the batteries are provided in Section IX, “Terminal Maintenance Procedures”.

After clearing the message (by pressing **[Return]**), you may then reconfigure the terminal as you desire.

### **No 'TO' device**

**Press RETURN to clear**

You attempted to initiate a device control data transfer (copy line, copy page, copy all) but no destination device is currently defined. Press **[Return]**, use the “**device control**” set of function keys to define an external printer as the “to” device, and then retry the copy operation.

**MEMORY FULL**

Press RETURN to clear

Memory Lock is enabled, and you have no more space available to enter more data in display memory.

**Function locked**

Press RETURN to clear

You attempted to use a function that has been locked-out. For example, after receiving the lock menu escape sequence (E<sub>C</sub>&q 1L) you tried to access a configuration menu or "modes" label. Also attempting to assign port 2 to the computer when this port is equipped with an 8-bit parallel module causes the error message to be displayed.

**Source = Destination**

Press RETURN to clear

You have defined the same device as the "from" device and the "to" device when copying data. The display is automatically the "from" device, and if it is the only device set as the "to" device the error message is displayed when you attempt to copy data to a device.

The following error messages exist on the HP 2394A terminal only.

**Illegal for edit type: ALPHABETIC**

Press RETURN to clear

With format mode enabled, you attempted to enter an illegal character into an "alphabetic" field.

**Illegal for edit type: ALPHANUMERIC**

Press RETURN to clear

With format mode enabled, you attempted to enter an illegal character into an "alphanumeric" field.

**Illegal for edit type: CONSTANT**

Press RETURN to clear

With format mode enabled, you attempted to alter a "constant" field.

**Illegal for edit type: DECIMAL**

Press RETURN to clear

With format mode enabled, you attempted to enter an illegal character into a "decimal" field.

**Illegal for edit type: IMPLIED DECIMAL**

Press RETURN to clear

With format mode enabled, you violated the format restrictions in an "implied decimal" field.

**Illegal for edit type: INTEGER**

Press RETURN to clear

With format mode enabled, you attempted to enter an illegal character into an "integer" field.

**Illegal for edit type: REQUIRED**

Press **RETURN** to clear

With format mode enabled, you attempted to transmit a for to the host computer (by press **[ENTER]**, for example) without having entered data into all “required” fields.

**Illegal for edit type: SIGNED DECIMAL**

Press **RETURN** to clear

With format mode enabled, you violated the format restrictions in a “signed decimal” field.

**Illegal for edit type: TOTAL FILL**

Press **RETURN** to clear

With format mode enabled, you violated the format restrictions in a “total fill” field.

## TERMINAL SELF-TESTS

The terminal includes five types of self-tests:

- Power-On Test
- Terminal Test
- Identify ROMs
- Port 1 Test
- Port 2 Test (if module present)

The Power-On test is automatically initiated as the result of a power-on sequence. All of the other tests must be initiated using the “**service keys**” (except the Terminal Test, which can also be initiated programmatically or by pressing **[f5]** in the **service keys** level of function keys).

### Power-on Test

The Power-On Test, which is performed automatically whenever you turn on the terminal’s power, does the following:

1. Tests the processor and verifies the integrity of all ROM (Read-Only Memory) and RAM (Random-Access Memory) chips within the terminal.

If the Power-On Test results are normal:

1. at power on, the terminal beeps once,
2. does the test for about 10 seconds,
3. beeps once again,
4. brings up the **MODES** group of softkey labels on the terminal screen.

If an error is found, one of the following will occur when the terminal is turned on:

- a. The terminal will fail to beep at all.
- b. The terminal will beep continuously.
- c. After the first beep, the terminal will beep several times and no softkey labels will appear on the screen.

If one of these error conditions occur, call the nearest HP Sales and Service Office and arrange to have the terminal repaired.

## Terminal Test

This test does the following:

1. Displays the message "TESTING" at the bottom of the screen (on the same line where the function key labels normally appear).
2. Verifies the integrity of all firmware ROM chips within the terminal.
3. Non-destructively verifies the integrity of all RAM chips within the terminal.
4. Displays the test pattern shown in Figure 8-1.

To initiate the Terminal test press the following keys in the sequence shown:

[User System] , [f3]  
service  
keys , [f5]  
TERMINAL  
TEST

Pressing [Shift] simultaneously with [Terminal test] will start a continuous test and an asterisk will appear in the label: [Terminal Test\*]. The only way to stop the continuous test is to press [Terminal Test\*] again.

If a ROM error is detected, the following message is displayed across the bottom of the screen:

**ROM ERR #x**  
**Press RETURN to clear**

where "x" will be the number 1 or 2 or 3 ROMs for the HP 2394A terminal. This message contains information identifying the bad ROM chip(s). In such a case, or for any other error message, write down the message so you can relate it accurately to your HP Service Representative over the telephone (this allows him to arrive prepared with the proper replacement parts).



If a RAM error is detected, the following message is displayed across the bottom of the screen:

**RAM ERR #x**  
Press RETURN to clear

where “x” is 1 to 4 in the HP 2392A terminal or “x” is 1 to 6 in the HP 2394A terminal. This message also contains information identifying the bad RAM chip(s). Write down the message and call your nearest HP Service representative.

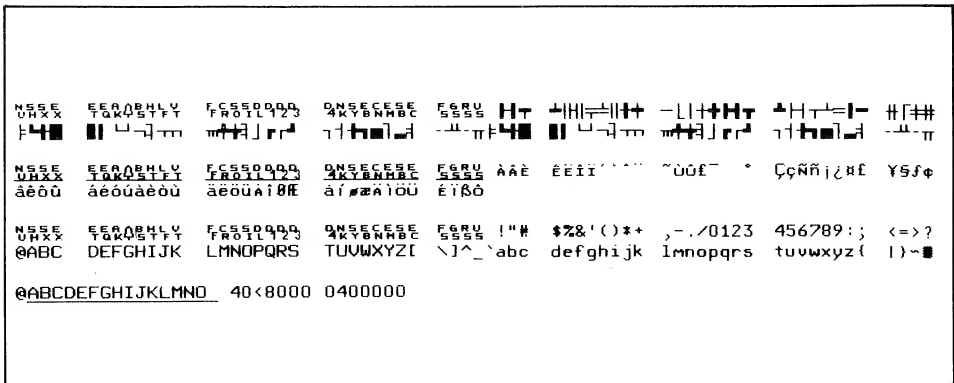


Figure 8-1. Screen Test Pattern (typical)

If the ROM and RAM chips all pass the test, but the test pattern on the screen is malformed, then this would suggest a problem with the video portion of the terminal (the sweep mechanism, the yoke alignment, and so forth).

## Identify ROMS

To generate a descriptive list of all ROM chips installed in the terminal, press the following keys in the order, shown:

[User System] , [f3]  
service  
keys , [f6]  
IDENTIFY  
ROMS

A list similar to the one shown in Figure 8-2 is displayed on the screen.

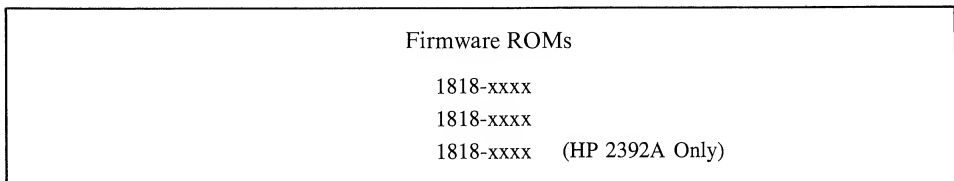


Figure 8-2. ROM Identification Listing

## Port 1 Test

The Port 1 test checks the 25-pin RS232C/HP422 port at the rear of the terminal. To enable the Port 1 self test, press the following keys in the sequence shown:

[User System] , 

[f3]
service keys

 , 

[f7]
PORT 1 TEST

and the test will be executed one time. Now, if you press [Shift] simultaneously with [PORT 1 TEST], then a continuous test will start and an asterisk will appear in the label [PORT 1 TEST\*]. The only way to stop this continuous test is to press [PORT 1 TEST] again (even a power down will not stop it).

A test hood (HP part no. 02620-60062 for RS232, 5061-3248 for HP422) must be connected to the port to perform the test, otherwise. "Datacomm Error 1" will appear on the screen.

The loopback test consist of a data loopback operation. While the test is running, "Testing" is displayed on the terminal screen. If no errors are found, the terminal beeps and displays the labels. If an error is found, an error message will appear on the terminal screen, similar to a ROM error.

The "Err #" field contains a numeric error code which is interpreted as follows:

- 1 = Test connector not present.
- 2 = Error in Control lines (RS232C only).
- 3 = Character did not loop back.
- 4 = Received character NOT same as one transmitted.
- 5 = Framing error in character.
- 6 = A character was overrun.

## Port 2 Test

Port 2 can be equipped with various modules (RS232C, 8-bit parallel).

When equipped with an RS232C module, the test hood should be 02620-60062.

When equipped with an 8-bit parallel module, no test can be performed. No action is taken when **[PORT 2 TEST]** is pressed.

To initiate PORT 2 press the following keys in the sequence shown:

[User System] , 

[f3]
User System

 , 

PORT 2 TEST
----------------

To start a continuous test press **[Shift] [PORT 2 TEST\*]** (an asterisk will appear in the label).

The only way to stop a continuous Port 2 test is to press **[PORT 2 TEST\*]** again.

(

(

(

# Terminal Maintenance Procedures

SECTION

IX

## INTRODUCTION

This section provides information on preventive maintenance for your terminal, such as cleaning the screen and keyboard.

## CLEANING THE SCREEN AND KEYBOARD

The display screen and the keyboard should be cleaned regularly to remove dust and grease. First, lightly dust the entire terminal using a damp, lint-free cloth or paper towel. The cloth or paper towel should be damp enough to pick up any dust, but should not be wet. Avoid wiping dust or lint into the key area of the keyboard.

Greasy smudges and fingerprints can be removed using most conventional spray cleaners. Avoid spraying between the keys.

DO NOT use petroleum-based cleaners (such as lighter fluid) or cleaners containing benzene, trichlorethylene, ammonia, dilute ammonia, or acetone because these chemicals could damage the terminal's plastic surfaces.

## BATTERY MAINTENANCE

The non-volatile portion of memory that contains the terminal's configuration data is protected against destruction by two batteries located just above the rear panel of your terminal. Figure 9-1 shows the rear panel and the location of the batteries.

The two batteries require no special care or maintenance. They should, however, be replaced every 12 months. You may purchase replacement batteries through conventional retail stores. When doing so, request UNION CARBIDE Type E90 or DURACELL MN9100. You may also order replacement batteries through your local HP Sales and Service Office using the following nomenclature and part number:

HP 2392A Battery  
HP Part No. 1420-0255

If your terminal has an external printer you may wish to record the various configuration menus on paper before removing the old batteries.

Replace the batteries as follows:

1. Switch ON the terminal.
2. Squeeze the tabs on either side of the battery support (shown in Figure 9-1) with enough pressure to disengage the flanges that hold the battery support in place.
3. Gently pull the support outward until it is completely free from the terminal housing.
4. Remove the old batteries from the support.
5. Install the new batteries in the support making sure that the positive end of the battery matches the positive end of the support (+ “to” + “and” – “to” –).
6. Reinsert the battery support into the terminal. The notch in the base of the support ensures it can only be inserted when correctly positioned. Check that the support engages with the terminal.

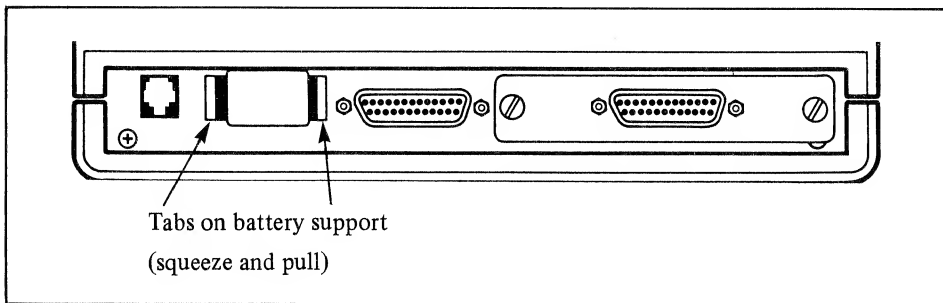


Figure 9-1. Removing the Batteries

# ANSI Operation

SECTION

X

## INTRODUCTION

The 2392A/2394A terminals can operate on computer systems using ANSI (American National Standards Institute) protocol. ANSI Operation allows the terminals to respond to ANSI escape sequences, adds functions to certain keys, and provides the terminals with answerback capability.

The following is a summary of the terminal's ANSI Operation features:

- 1) Hewlett-Packard (HP) mode of operation: the terminal operates as a standard HP 2392A/2394A terminal. It uses the capabilities, communications and escape sequences described in the 2392A/2394A Reference Manual.
- 2) ANSI mode of operation: the terminal recognized and executes a sub-set of the terminal escape sequences specified in the American National Standards Institute documents X3.41-1974 and X3.64-1979. It also recognizes and executes some of the standard HP parameterized escape sequences. In ANSI mode:
  - a) An ANSI Configuration Menu allows the user to define:
    - The terminal "answer back" message (response to a computer enquiry, Enq).
    - Tab settings, which are saved in the terminal's non-volatile memory.
    - The [Back space] key operation (the key can be used as a backspace or a delete key).
    - The size of display memory.
  - b) Keys [◀], [▶], [▲], [▼] can be programmed to operate as cursor control keys or "application mode" keys.
  - c) The function of some of the keys in the numeric pad is changed. A "numeric pad overlay", supplied with the terminal, indicates the function of these keys. Also, the keys in the numeric pad can be programmed to operate as numeric keys or "application mode" keys.
  - d) A terminal mode indicator ("ANSI") and four programmable application flags are available in the Status Line.
  - e) Two line drawing sets are available.
- 3) EM52 mode: the terminal emulates a VT52<sup>\*</sup> compatible terminal by recognizing and executing escape sequences written for a DEC VT52<sup>\*</sup> terminal.

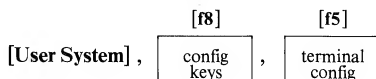
In EM52 mode, the ANSI Configuration Menu is available to: select an "answer back" message, define tab settings and define the [Back space] key operation. The function of some of the keys in the numeric pad is changed (the same as for ANSI mode). Also, the keys in the numeric pad can be programmed to operate as numeric keys or "applications mode" keys. A terminal indicator ("EM52") is displayed in the Status Line. Note that Status Line flags are not available, and keys [◀], [▶], [▲] and [▼] can only operate as cursor control keys.

<sup>\*</sup> VT52 and DEC are registered trademarks of Digital Equipment Corporation.

## ANSI/EM52/HP MODE CONFIGURATION

When accessing the ANSI/EM52 modes, the field labeled “TermMode”, can be set to “HP” (default), “EM52” or “ANSI”. The setting of this field determines the terminal’s mode of operation. As with the rest of this menu, the contents of the field are saved in non-volatile memory.

The “TermMode” field is configured in the same manner as the other fields in this menu. To alter the setting, display the Terminal Configuration Menu by pressing the keys in the following order:



This displays the menu shown in Figure 10-1.

Then place the cursor in the field with either the [TAB] key or the cursor keys. Use the **NEXT CHOICE** or **PREVIOUS CHOICE** function keys to change the field to the desired choice. To save the new configuration, press the **SAVE CONFIG** function key. The saved configuration becomes the active configuration, and is saved in non-volatile memory. If the terminal is set to EM52 or ANSI mode, a terminal mode indicator is displayed in the Status Line. The cursor is homed up and the display cleared when a new terminal mode is saved.

The default setting for the “TermMode” field is HP. Figures 10-1 and 10-2 show the Terminal Configuration Menus with the default settings. For information on the other terminal configuration parameters, plus datacomm and external device configuration parameters, refer to Sections II, V, and VI of the 2392A/2394A Reference Manual.

Note that in ANSI/EM52 Mode, the terminal is automatically set to Character Mode (asterisk removed from **BLOCK MODE** label), and the associated function key is disabled. All Block Mode configuration menu fields are ignored.

TERMINAL CONFIGURATION							
Datacomm/ExtDev	PORT1/PORT2		Keyboard	USASCII			
TerminalId	2392A		Language	ENGLISH			
LocalEcho	OFF	CapsLock	OFF	Start Col	01	Bell	ON
XmitFnctn (A)	NO	Spow (B)	NO	InhEolWrp (C)	NO	Line/Page (D)	LINE
InhHndShk (G)	NO	Inh DC2 (H)	NO	EscXfer (N)	NO		
FldSeparator	U <sub>s</sub>	BlkTerminator	R <sub>s</sub>	ReturnDef	C <sub>r</sub>	TermMode	HP
Note: “Datacomm/ExtDev” is only displayed when the terminal has two ports.							
Terminal Configuration Menu Labels							
SAVE CONFIG	NEXT CHOICE	PREVIOUS CHOICE	DEFAULT VALUES			DISPLAY FUNCTIONS	config keys
[f1]	[f2]	[f3]	[f4]	[f5]	[f6]	[f7]	[f8]

Figure 10-1. HP 2392A Terminal Configuration Menu (default values)



TERMINAL CONFIGURATION							
Datacomm/ExtDev		<input type="text" value="PORT1/PORT2"/>		Keyboard		<input type="text" value="USASCII"/>	
Terminal ID		<input type="text" value="2394A"/>		Language		<input type="text" value="ENGLISH"/>	
Local Echo	<input type="text" value="OFF"/>	CapsLock	<input type="text" value="OFF"/>	Start Col	<input type="text" value="01"/>	Bell	<input type="text" value="NO"/>
XmitFnctn (A)	<input type="text" value="NO"/>	SPOW (B)	<input type="text" value="NO"/>	InhEolWrp (C)	<input type="text" value="NO"/>	Line/Page (D)	<input type="text" value="LINE"/>
InhHndShk (G)	<input type="text" value="NO"/>	Inh DC2 (H)	<input type="text" value="NO"/>			Esc Xfer (N)	<input type="text" value="NO"/>
						FormsBufSize (256x)	<input type="text" value="0"/>
FldSeparator	<input type="text"/>	BlkTerminator	<input type="text"/>	Return = Enter	<input type="text" value="NO"/>	ReturnDef	<input type="text"/>
Tab = Spaces	<input type="text" value="NO"/>					TermMode	<input type="text" value="HP"/>
FORMAT MODE							
Decimal Type	<input type="text" value="US"/>	Imp Dec Digits	<input type="text" value="2"/>	Transmit	<input type="text" value="All Fields"/>	Print	<input type="text" value="Fields"/>
Note: "Datacomm/ExtDev" is only displayed when the terminal has two ports.							
Terminal Configuration Menu Labels							
<input type="button" value="SAVE CONFIG"/>	<input type="button" value="NEXT CHOICE"/>	<input type="button" value="PREVIOUS CHOICE"/>	<input type="button" value="DEFAULT VALUES"/>	<input type="button" value="POWER ON VALUES"/>	<input type="button" value="ACTIVE VALUES"/>	<input type="button" value="DISPLAY FUNCTIONS"/>	<input type="button" value="config keys"/>

Figure 10-2. HP 2394A Terminal Configuration Menu (default values)

## MODES OF OPERATION

ANSI Operation has three operating modes: HP, EM52, and ANSI. At power-on, the mode is determined by the setting in the Terminal Configuration Menu. After power-on, the mode can be changed programmatically using escape sequences, or by reconfiguring the terminal. Note that programmatically changing the mode of operation does not alter the contents of the non-volatile memory (which sets the operating mode at power-on or after a hard reset).

### HP Mode

In HP mode, the terminal performs all the functions of a standard 2392A/2394A terminal. It responds to the escape sequences and performs all the terminal functions described in Sections 1 to 9 and Appendices A and B of the 2392A/2394A Reference Manual. The terminal does not respond to either VT52® or ANSI escape sequences.

## EM52 Mode

In EM52 mode, the terminal emulates a VT52® compatible terminal. The terminal responds to the escape sequences written for a DEC VT52® terminal (listed at the end of this section). The terminal does not respond to either HP or ANSI escape sequences.

In EM52 mode, a terminal mode indicator “EM52” is displayed in the Status Line, and the terminal provides some of the features available in ANSI mode:

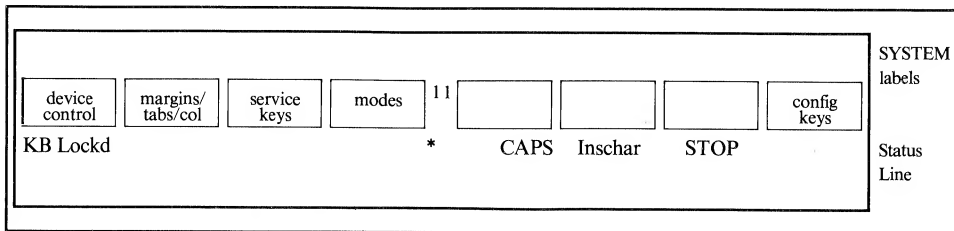


Figure 10-3. HP Mode Screen Status Line

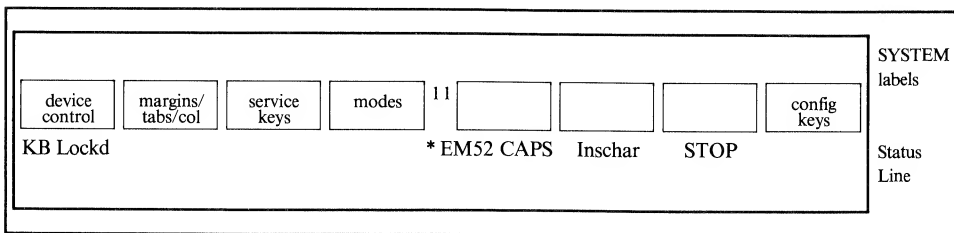


Figure 10-4. EM52 Mode Screen Status Line

1) The function of some of the keys in the numeric pad is changed:

- a) Keys **[\*]**, **[/]**, **[+]** and **[-]** act as program function keys **[PF1]** to **[PF4]**.
- b) **[Enter]** acts as **[-]**.
- c) **[Tab]** acts as **[Enter]**.
- d) The unmarked key above the “\*” acts as the linefeed key.

The “numeric pad overlay”, supplied with the terminal, indicates the function of these keys.

The keys in the numeric pad (except **[PF1]** to **[PF4]**) can be programmed to operate as numeric pad keys or “applications mode” keys.

2) The terminal may use any of the parameters configurable in the ANSI Configuration Menu. These are:

- a) Answerback message (up to 40 characters).
- b) Size of display memory: one page or multiple pages.
- c) [Back space] key operation (backspace/delete character).
- d) Tab stops (which are saved in non-volatile memory).

These features are described later in this section.

Note that, in EM52 mode, the four application flags in the Status Line are not available, keys [◀], [▶], [▲] and [▼] can only be used as cursor control keys, and the [Insert char], [Delete char], [Insert line], and [Delete line] keys are disabled.

## ANSI Mode

ANSI is the mode of operation that gives the terminal the ability to operate with a computer system using ANSI control sequences. In ANSI mode, the terminal responds to the ANSI escape sequences described at the end of this section. It also responds to some of the HP parameterized escape sequences recognized by the standard terminal. It does not respond to VT52® escape sequences.

In ANSI mode, a terminal mode indicator “ANSI” is displayed in the Status Line. Four application flags (L1, L2, L3 and L4) in the Status Line can be turned on and off programmatically. The function of some of the keys in the numeric pad is changed:

- Keys [∗], [/], [+], and [−] act as program function keys [PF1] to [PF4].
- [Enter] acts as [−], and [Tab] as [Enter].
- The unmarked key above the “∗” acts as the linefeed key.

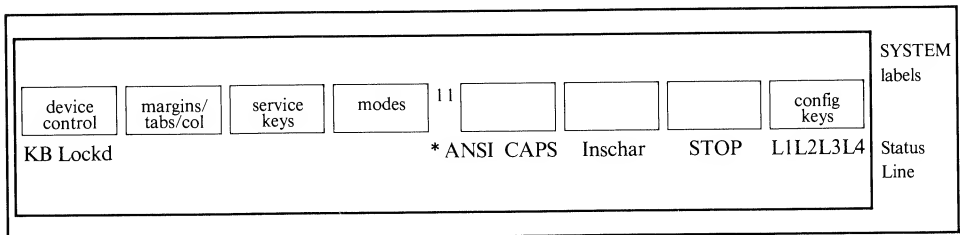


Figure 10-4. ANSI Mode Screen Status Line

The “numeric pad overlay”, supplied with the terminal, indicates the function of these keys. The same overlay is used for both ANSI and EM52 mode.

The keys in the numeric pad (except for [PF1] to [PF4]) can be programmed to operate as numeric pad keys or “applications mode” keys. Keys [◀], [▶], [▲], [▼] can be programmed to operate as cursor control keys or “application mode” keys.

The [Insert char], [Delete char], [Insert line], and [Delete line] keys are disabled in ANSI Mode.

Two line drawing character sets are available: one is the same as that used in HP mode, the other uses elements different from those in HP mode.

In ANSI mode, the terminal can transmit a user-definable answerback message by pressing the [CTRL] and [BREAK] keys, together. Also, a long break (3.5 seconds) can be transmitted to the host computer by pressing the [CTRL] and [STOP] keys together.

In addition, the terminal may use any of the parameters in the ANSI Configuration Menu. These are:

- Answerback message (up to 40 characters).
- Size of display memory: one page or multiple pages.
- [Back space] key operation (backspace/delete character).
- Tab stops (which are saved in non-volatile memory).

These features are detailed later in this section.

## ESCAPE SEQUENCES

The terminal recognizes different escape sequences according to the operating mode. If the mode is changed, any previously received escape sequences that are not recognized in the new mode are lost. For instance, if the terminal was in ANSI mode with a scrolling region defined, and the mode is changed to HP, the scrolling is ignored (the scrolling region is NOT re-instated if the terminal is subsequently re-set to ANSI mode).

### HP Mode

When the terminal is configured for HP mode, it responds to the escape sequences listed in Appendix A of the 2392A Reference Manual. The terminal does not respond to either VT52® or ANSI escape sequences.

### EM52 Mode

When the terminal is configured for EM52 mode, it responds to the escape sequences written for a DEC VT52® terminal listed at the end of this section. The terminal does not respond to either HP or ANSI escape sequences.

### ANSI Mode

When the terminal is configured for ANSI mode, it responds to the ANSI compatible escape sequences found at the end of this section. It also responds to the following HP parameterized escape sequences recognized by the standard terminal: “Ec&f”, “Ec&j”, “Ec&k”, “Ec&p”, “Ec&q”, “Ec&s”, “Ec&w”, and “Ec&K”.

These HP escape sequences are described in Appendix A of the 2392A/2394A Reference Manual. The terminal does not respond to VT52® escape sequences.

The ANSI sequences are a subset of those specified in Standards ANSI X3.41-1974 and ANSI X3.64-1979. Those sequences designated as private by ANSI and ISO code extension standards (ANSI X3.41-1974 and ISO 2022 1973, respectively) are not yet standardized. Therefore, Hewlett-Packard has assigned functions to some of these ANSI compatible sequences, and prefixed their mnemonic with “HP”. All the other mnemonics used are the same as those specified in the ANSI standards.

## Escape Sequences Generated By The Keyboard

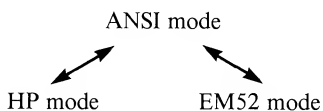
Escape sequences are also generated at the keyboard by active edit keys (such as [Clear line] and [Clear display]), system-level function keys (such as “Memory Lock”), and cursor control keys. The transmission of these escape sequences from the terminal to the host computer depends on the terminal mode:

- In HP mode, the escape sequences are only transmitted to the host computer when the “XmitFncn(A)” field (in the Terminal Configuration Menu, see Figure 10-1) is set to “YES”.
- In ANSI mode, escape sequences generated by both the numeric keypad keys (when set to application mode) and the cursor control keys are always transmitted, and the “XmitFncn(A)” field is ignored. For the remaining keys, the escape sequences are only transmitted when the “XmitFncn(A)” field is set to “YES”.
- In EM52 mode, only escape sequences generated by the numeric keypad keys (when set to application mode) and the cursor control keys are transmitted. The “XmitFncn(A)” field is ignored.

Non-transmitted escape sequences are executed locally (at the terminal). Transmitted escape sequences are only executed locally if the computer performs a remote echo or the terminal’s local echo is ON.

## Selecting Terminal Mode Programmatically

The terminal mode can be configured programmatically from the host computer, but only in the following directions:



The escape sequences used to configure the mode are as follows:

- 1) HP mode. HP mode can only be selected when currently operating in ANSI mode, using ANSI escape sequence:

`ESC &k0\` (from ANSI mode to HP mode)

- 2) EM52 mode. EM52 mode can only be selected when currently operating in ANSI mode, using ANSI private escape sequence:

$E_C[?21$  (from ANSI mode to EM52 mode)

- 3) ANSI mode. ANSI mode can be selected when operating in either HP or EM52 mode using:

a) HP escape sequence:  $E_C \&k1 \backslash$  (from HP mode to ANSI mode)

b) EM52 escape sequence:  $E_C <$  (from EM52 mode to ANSI mode)

## CURSOR CONTROL KEYS

In HP and EM52 mode, the four keys [ $\blacktriangleleft$ ], [ $\blacktriangleright$ ], [ $\blacktriangleup$ ] and [ $\blacktriangledown$ ] operate as cursor control keys, and generate the characters listed in Table 10-1. In ANSI mode (only), the four keys [ $\blacktriangleleft$ ], [ $\blacktriangleright$ ], [ $\blacktriangleup$ ] and [ $\blacktriangledown$ ] can be operated as:

- 1) Cursor control keys. These keys generate control sequences that control the movement of the cursor (see Table 10-2).
- 2) "Application mode" keys. These keys generate different control sequences (see Table 10-2), the functions performed depends on the application program.

The cursor key application mode is set or reset programmatically using the following ANSI escape sequences:

SET (= application mode):  $E_C[?1h$

RESET (= cursor control mode):  $E_C[?1l$

Tables 10-1 and 10-2 list the control sequences generated by each key. Note that:

- In Local Mode, control sequences are processed locally.
- In Remote Mode, control sequences are always transmitted. An echo must be performed for the control sequences to be processed by the terminal. For this, either the host must perform a remote echo or the terminal's Local Echo parameter must be set to "ON".

The keys revert to cursor control keys at power-on or after a hard reset.

Table 10-1. HP And EM52 Control Sequences generated By Cursor Keys

Key	HP Mode	EM52 Mode
[ $\blacktriangleup$ ] Cursor up	$E_C A$	$E_C A$
[ $\blacktriangledown$ ] Cursor down	$E_C B$	$E_C B$
[ $\blacktriangleright$ ] Cursor right	$E_C C$	$E_C C$
[ $\blacktriangleleft$ ] Cursor left	$E_C D$	$E_C D$

Table 10-2. ANSI Control Sequences Generated By Cursor Keys

Key	Cursor Key Application Mode RESET (cursor movement keys)		Cursor Key Application Mode SET (application mode keys)
[▲]	$E_C[A$	Cursor up	$E_C \&0A$
[▼]	$E_C[B$	Cursor down	$E_C \&0A$
[▶]	$E_C[C$	Cursor right	$E_C \&0C$
[◀]	$E_C[D$	Cursor left	$E_C \&0D$

Application function

## NUMERIC KEYPAD

When the terminal is set for ANSI or EM52 operations, the numeric pad is “re-mapped” with:

- Keys [\*, [/], [+], and [-] used as program function keys [PF1] to [PF4].
- [Enter] as [-].
- [Tab] as [Enter].
- Keys [0] to [9] are unchanged.
- The unmarked key above the “\*” as linefeed key.

The “numeric pad overlay”, supplied with the terminal, indicates the function of these keys. The same overlay is used for both ANSI and EM52 mode. See Figure 10-5.

The numeric keypad has two modes of operation: numeric mode and application mode.

In numeric mode, the keys operate as re-mapped above. In other words, keys in the numeric keypad (except [PF1] to [PF4]) transmit the same codes as the corresponding keys on the main keyboard (see Table 10-3). Therefore, the host computer cannot tell if these keys were pressed on the numeric keypad or on the main keyboard.

In application mode, the keys in the numeric pad (except [PF1] to [PF4]) generate different control sequences (see Table 10-3), the interpretation of which depends on the application program.

Function keys [PF1] to [PF4] generate the same control sequences in both numeric and application mode (see Table 10-3). The use of these keys depends on the application program.

The numeric keypad application mode is set or reset programmatically using the following escape sequences:

	ANSI Mode	EM52 Mode
SET (= application mode):	$E_C=$	$E_C=$
RESET (= numeric mode):	$E_C>$	$E_C>$

Table 10-3 lists the control sequences generated by each key.

The keypad reverts to numeric mode at power-on or after a hard reset.

## ALTERNATE CHARACTER SETS

### HP Mode

In HP mode, the terminal has two character sets: the base set and the alternate set. These two sets are fixed (and cannot be changed from the host or from the keyboard). The base set is the configured keyboard type and the alternate set is the HP line drawing set.

Table 10-3. ANSI And EM52 Control Sequences Generated By Numeric Keypad

Numeric Keypad Overlay Legend	CONTROL SEQUENCE SENT TO HOST COMPUTER			
	ANSI		EM52	
	Numeric mode	Application mode	Numeric mode	Application mode
0	0	$\text{E}_C \text{ O p}$	0	$\text{E}_C ? \text{ p}$
1	1	$\text{E}_C \text{ O q}$	1	$\text{E}_C ? \text{ q}$
2	2	$\text{E}_C \text{ O r}$	2	$\text{E}_C ? \text{ r}$
3	3	$\text{E}_C \text{ O s}$	3	$\text{E}_C ? \text{ s}$
4	4	$\text{E}_C \text{ O t}$	4	$\text{E}_C ? \text{ t}$
5	5	$\text{E}_C \text{ O u}$	5	$\text{E}_C ? \text{ u}$
6	6	$\text{E}_C \text{ O v}$	6	$\text{E}_C ? \text{ v}$
7	7	$\text{E}_C \text{ O w}$	7	$\text{E}_C ? \text{ w}$
8	8	$\text{E}_C \text{ O x}$	8	$\text{E}_C ? \text{ x}$
9	9	$\text{E}_C \text{ O y}$	9	$\text{E}_C ? \text{ y}$
–	–	$\text{E}_C \text{ O m}$	–	$\text{E}_C ? \text{ m}$
.	.	$\text{E}_C \text{ O n}$	.	$\text{E}_C ? \text{ n}$
,	,	$\text{E}_C \text{ O l}$	,	$\text{E}_C ? \text{ l}$
[Enter]	Same as [Return]	$\text{E}_C \text{ O M}$	Same as [Return]	$\text{E}_C ? \text{ M}$
PF1	$\text{E}_C \text{ O P}$	$\text{E}_C \text{ O P}$	$\text{E}_C \text{ P}$	$\text{E}_C \text{ P}$
PF2	$\text{E}_C \text{ O Q}$	$\text{E}_C \text{ O Q}$	$\text{E}_C \text{ Q}$	$\text{E}_C \text{ Q}$
PF3	$\text{E}_C \text{ O R}$	$\text{E}_C \text{ O R}$	$\text{E}_C \text{ R}$	$\text{E}_C \text{ R}$
PF4	$\text{E}_C \text{ O S}$	$\text{E}_C \text{ O S}$	$\text{E}_C \text{ S}$	$\text{E}_C \text{ S}$



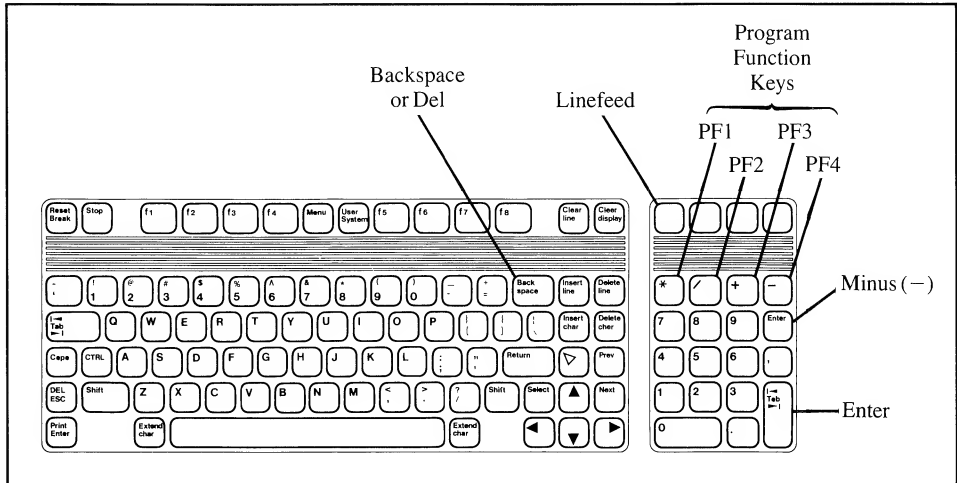


Figure 10-5. Numeric Pad Overlay

The base set is enabled by default at power-on or by receipt of a Shift-in (S<sub>i</sub>) control character (generated at the keyboard using keys [CTRL] [O]).

The alternate set is enabled by the Shift-out (S<sub>o</sub>) control character [CTRL] [N].

## EM52 Mode

In EM52 mode, the terminal only has one character set, which is the configured keyboard type.

## ANSI Mode

In ANSI mode, the terminal has two character sets: base and alternate. These sets are selectable from the host or keyboard, using ANSI escape sequence:

- $E_C$  ( Ps – select base character set (G0)
- $E_C$  ) Ps – select alternate character set (G1)

Where Ps can have the values:

- B = configured language.
- 0 = ANSI line drawing set.
- 3 = HP line drawing set.

At power-on, the default base character set (G0) is the configured language set, and the default alternate character set (G1) is also the configured language set.

To invoke the base set (previously selected by the appropriate ANSI escape sequence) while operating in the alternate set, issue a Shift-in (S<sub>I</sub>) control character.

The alternate set (G1) is enabled by the Shift-out (S<sub>O</sub>) control character. The alternate set stays enabled until you perform a hard reset or send a Shift-in (S<sub>I</sub>) control character.

Line Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Encoding Character	h	t	+	h	h	h	h	h	h	h	h	h	h	h	h	h	h	h	h	h	h	h	h	h	h	h	h	h	h	h	h	
Corresponding USASCII Character	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	!	@	#	\$	%	&
	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	^	1	2	3	4	5
																										6	7	8	9	0	-	=
																										[	]	<	>	;	,	/

Figure 10-6. HP Line Drawing Set Elements (USASCII Keyboard)




































































Line Drawing Character	                        
Keyboard Character	a b c d e f g h i j k l m n o p q r s t u v w x y z
Line Drawing Character	                    
Keyboard Character	` 1 2 3 4 5 6 7 8 9 0 - = [ ] \ ; ' , . /
Line Drawing Character	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Keyboard Character	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Line Drawing Character	                    
Keyboard Character	~ ! @ # \$ % ^ & * ( ) _ + < >   : " < > ?

Figure 10-7. ANSI Line Drawing Set Elements (USASCII Keyboard)

## ANSI CONFIGURATION MENU

When the terminal is set for ANSI Mode or EM52 Mode, the ANSI Configuration Menu is available to define:

- 1) The size of the display memory.
- 2) The terminal “answer back” message (response to a computer enquiry, Enq).
- 3) Tab settings, which are saved in the terminal’s non-volatile memory.
- 4) The **[Back space]** key operation, which can be used as both a backspace and a delete key.

ANSI CONFIGURATION	
MultiPage	[NO]
	BackspaceDef (Unshift/shft) [Del/Backspace]
Answerback Message	
<div style="display: flex; justify-content: space-around; font-size: small;"> <span>T</span><span>T</span><span>T</span><span>T</span><span>T</span><span>T</span><span>T</span><span>T</span><span>T</span> </div> <div style="display: flex; justify-content: space-around; font-size: small;"> <span>1234567890123456789012345678901234567890123456789012345678901234567890</span> </div>	
Note: "Answerback Message" is only displayed when key [f5] (<ANSWER BACK>) is pressed.	
ANSI Configuration Menu Labels	
<div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 2px 5px; font-size: x-small;">SAVE CONFIG</div> <div style="border: 1px solid black; padding: 2px 5px; font-size: x-small;">NEXT CHOICE</div> <div style="border: 1px solid black; padding: 2px 5px; font-size: x-small;">PREVIOUS CHOICE</div> <div style="border: 1px solid black; padding: 2px 5px; font-size: x-small;">DEFAULT VALUES</div> <div style="border: 1px solid black; padding: 2px 5px; font-size: x-small;">ANSWER BACK</div> <div style="border: 1px solid black; padding: 2px 5px; font-size: x-small;">CLR ALL TABS</div> <div style="border: 1px solid black; padding: 2px 5px; font-size: x-small;">DISPLAY FUNCTNS</div> <div style="border: 1px solid black; padding: 2px 5px; font-size: x-small;">config keys</div> </div>	
LABEL	FUNCTION (ANSI MENU CONTROL KEYS)
<SAVE CONFIG	Saves the displayed configuration parameters in the non-volatile memory, and returns terminal to normal operations with SYSTEM labels.
<NEXT CHOICE>	When the cursor is in the "MultiPage" or "Backspace def" field or "Tab stop" line, the <NEXT CHOICE> and <PREVIOUS CHOICE> keys scroll forward/backward through the pre-set list of values.
<PREVIOUS CHOICE>	
<DEFAULT VALUES>	Sets all displayed parameter fields to their pre-set default values (regardless of the cursor position), and moves cursor to the first parameter field. The Answerback Message is only reset (to no message) when Answerback field is displayed.
<ANSWER BACK>	Toggles ON/OFF "Answerback Message" field. When ON, allows data to be entered in field. Note that a configured Answerback Message cannot be recalled from non-volatile memory.
<CLR ALL TABS>	Clears all tab stops in menu (in all columns, except implicit stop at column 1).
<DISPLAY FUNCTNS*>	When cursor is in "Answerback Message" field, allows control characters to be entered and displayed symbolically in field.
<config keys>	Terminates the configuration mode without saving the entered values, and returns the terminal to normal operations with main configuration labels.

Figure 10-8. ANSI Configuration Menu (default values)

To display the ANSI Configuration Menu, first ensure that the terminal is in ANSI or EM52 mode, then press the keys in the following order:

[f8]                      [f5]  
 [User System] , config  
keys , ansi  
config

This displays the menu shown in Figure 10-8. The parameters in this menu can be configured as described below.

## Multipage Field

The 2392A is a multipage terminal that dynamically stores data. The standard terminal typically stores up to 4 pages (96 lines) of text in display memory. Applications programs designed for ANSI and EM52 terminals, however, often require the terminal to have only a single page of memory (24 lines). Consequently, the 2392A can be configured to use a single page or multiple pages. When a single page is used, keys **[Next]**, **[Prev]**, roll up (**[Shift]** **[▲]**) and roll down (**[Shift]** **[▼]**) are disabled.

To change the “MultiPage” parameter in the ANSI Configuration Menu:

- 1) Place the cursor in the field with either the **[Tab]** key or the cursor control keys. Use the **NEXT CHOICE** or **PREVIOUS CHOICE** function keys to change the field to the desired choice.
- 2) Save the new configuration by pressing the **SAVE CONFIG** function key. This causes the **SYSTEM** labels to be displayed. The saved configuration becomes the active configuration, and is saved in non-volatile memory.

In ANSI mode, the multipage parameter can be set and reset programmatically using the following ANSI escape sequences:

SET (= multipage):  $\text{Fc} [> 1 \text{h}]$

RESET (= single page):  $\text{Fc} [> 1 \text{l}]$

Changing the value of this parameter will clear display memory and home up the cursor. The multipage parameter value set programmatically is not saved in the non-volatile memory. If the power is switched OFF, the terminal reverts to the value in the “Multipage” field.

## Backspace Definition Field

In ANSI/EM52 mode, the **[Back space]** key has two functions:

- Backspace – generates a  $\text{B}_\text{s}$  control code (hexadecimal 08) which moves the cursor one character backwards (leftwards) along the line, stopping at column 1. This is a non-destructive backspace function.
- Delete – generates a DEL control code (hexadecimal 7F) which is usually interpreted by the host computer to mean “delete the preceding character and move the cursor one space left”.

The “Backspace def” field in the ANSI Configuration Menu specifies the operation of the **[Back space]** key when pressed by itself (unshifted) and when pressed with **[Shift]** (shifted). To change the “Backspace def” parameter:

- 1) Place the cursor in the field with either the **[Tab]** key or the cursor control keys. Use the **NEXT CHOICE** or **PREVIOUS CHOICE** function keys to change the field to the desired choice (backspace/delete or delete/backspace).

- 2) Save the new configuration by pressing the **SAVE CONFIG** function key. This causes the **SYSTEM** labels to be displayed. The saved configuration becomes the active configuration, and is saved in non-volatile memory.

The operation of the **[Back space]** key is not programmable from the host computer.

## Setting And Saving Tabs

Tab stops defined in the ANSI Configuration Menu are saved in non-volatile memory, and are not lost if the terminal is powered off. Note that these tab stops are only available in ANSI/EM52 mode; they are ignored in HP mode. To set ANSI tab stops:

- 1) Place the cursor in the desired position in the tab stop indicator line (at the bottom of the menu). Then use the **NEXT CHOICE** or **PREVIOUS CHOICE** key to toggle on and off the tab stop (a stop is indicated by a “T”). The label **CLR ALL TABS** clears all the tab stops in all columns in the menu (except for the implicit tab at column 1).
- 2) Press key **[f1]** (**SAVE CONFIG**) to store the tab stops, and return the display to the **SYSTEM** labels. The saved tab stops become the active tab stops and are saved in non-volatile memory.

Tab stops can also be set using the **SET TAB** screen label (accessed through the **margins/tabs/col** label). However, these tab stops are not retained in non-volatile memory.

In HP and ANSI mode, tab stops can be set and cleared programmatically using the following escape sequences (in EM52 mode, tab stops cannot be set/cleared programmatically):

	HP Mode	ANSI Mode
SET tab stop:	$\text{E}_C1$	$\text{E}_CH$
CLEAR tab stop:	$\text{E}_C2$	$\text{E}_C[0g$ or $\text{E}_C[g$
CLEAR ALL tab stops:	$\text{E}_C3$	$\text{E}_C[3g$

The tab stops are not saved in the non-volatile memory. If the power is switched OFF (or a hard reset is performed), the terminal reverts to the values in non-volatile memory.

## Answerback Message Field

In ANSI/EM52 mode, the terminal has the ability to identify itself to the host computer with a user-defined character string called an answerback message. The answerback message is transmitted when the terminal receives an “Enq” character (hexadecimal 05) from the computer.

When the ANSI Configuration Menu is first displayed, the “Answerback Message” field does not appear. This field is only displayed when the **ANSWER BACK** key is pressed. In addition (for security), the configured answerback message cannot be recalled once it has been defined.

When the ANSI Configuration Menu is first displayed, the "Answerback Message" field does not appear. This field is only displayed when the **ANSWER BACK** key is pressed. In addition (for security), the configured answerback message cannot be recalled from non-volatile memory.

By default no message exists, and it is up to the user to define the message. This can be done as follows:

- 1) Press key **[f5]** (**ANSWER BACK**) to display the "Answerback Message" parameter. The message field (displayed in half-bright inverse video) is blank, with the cursor at the start of the field. You can then enter a message of up to 40 characters in the field.

Each entered character (including trailing blanks) appears in full-bright inverse video. The unused portion of the field (right-hand side) remains in half-bright inverse video. When the message is transmitted, only the portion in full-bright inverse video is sent. The entered message can be edited using **[Clear line]** (which clears to the end of the field), **[Delete char]** and **[Insert char]** keys. The function key labeled **DISPLAY FUNCTIONS** allows the message to contain terminal control characters (displayed symbolically).

- 2) Ensure the answerback message field is displayed, then press **SAVE CONFIG** (key **[f1]**). This causes the **SYSTEM** labels to be redisplayed and all the displayed menu parameters to be saved. The saved answerback message becomes the active message, and is saved in non-volatile memory (replacing the previously stored message). If the answerback message is not displayed when **SAVE CONFIG** is pressed, the previously stored message is retained.

Note that the answerback message is not programmable from the host computer. The answerback message can be triggered from the keyboard by pressing the **[CTRL]** and **[BREAK]** keys, together.

## ANSI CONTROL SEQUENCE SUMMARY

The control sequences recognized and processed by the terminal when in ANSI mode are detailed below.

**Terms.** The following terms apply in ANSI mode:

- |                                    |   |
|------------------------------------|---|
| Active position:                   | The position of the cursor.   |
| Control sequence:                  | <p>A string of characters that is used to perform a special function, that contains the Control Sequence Introducer (<math>\text{ESC}</math>), some parameters which may be optional, and a final character.</p> <p>If a sequence contains several selective or numeric parameters, successive parameters must be separated by semicolons ";" (hexadecimal 3B).</p>   |
| Numeric Parameter ( $\text{Pn}$ ): | <p>A string of numbers which represents a numeric value between two range limits.</p> <p>For example, when moving the cursor with a CUF (Cursor Forward) sequence: <math>\text{ESC} [\text{Pn} \text{C}</math>.</p> <p><math>\text{Pn}</math> is a numeric parameter that can have any value between 1 and 80. If the cursor is to be moved 7 places to the right, the sequence is: <math>\text{ESC} [ 7 \text{C}</math>.</p> |

- Scrolling region: The part of the display which rolls up when new data is received by the terminal. If no top or bottom margins are in effect, the scrolling region is the whole display.
- Selective parameter (Ps): A string of characters which selects one function among a list of several functions. The selective parameter, designated as Ps, may ONLY have a value which is chosen from the list of functions. Any other value will be interpreted as an error.

## Control Characters

Control characters (hex 00 to 1F inclusive) are excluded from the control sequence syntax, but may be embedded within a control sequence. Embedded control characters are executed as soon as they are received by the terminal, provided display functions mode is not set. The processing of the control sequence then continues with the next character encountered.

Table 10-4 lists the control characters processed by the terminal in ANSI or EM52 Mode. For a list of the control characters processed in HP mode, refer to Appendix A of the 2392A Reference Manual.

Table 10-4. ANSI/EM52 Mode Control Character Recognition

Control Character	Hex Code	Terminal Action
NUL	00	Ignored on input.
ENQ	05	Transmits an answerback message.
BEL	07	Sounds the bell.
BS	08	Moves the cursor one position to the left, stopping at column 1.
HT	09	Moves the cursor to the next tab stop, stopping at column 80 if no further tab stops are in the line.
LF	0A	Executes a linefeed or a new line operation (see new line mode, LNM).
VT	0B	Interpreted as LF.
FF	0C	Interpreted as LF.
CR	0A	Moves cursor to column 1 of current line.
SO	0E	Invokes G1 character set, as defined by SCS sequence (see select character set sequence).
SI	0F	Invokes G0 character set, as defined by SCS sequence.
X-ON	11	Causes terminal to resume transmission.
X-OFF	13	Causes terminal to stop transmitting all characters except X-ON and X-OFF.
CAN	18	If sent during a control sequence, the sequence is immediately terminated and not executed. It also causes the error character "■" to be displayed.
SUB	1A	Interpreted as CAN.
EC	1B	Introduces a control sequence.
DEL	7F	Ignored on input.



## ANSI Cursor Control Sequences

The following cursor control sequences are responded to or sent by the terminal in ANSI mode:

	Mnemonic	Description
ANSI specified sequences	CBT	Cursor Back Tab
	CHA	Cursor Horizontal Absolute
	CNL	Cursor Next Line
	CPL	Cursor Preceding Line
	CPR	Cursor Position Report
	CUB	Cursor Backward
	CUD	Cursor Down
	CUF	Cursor Forward
	CUP	Cursor Position
	CUU	Cursor Up
	HTS	Horizontal Tab Set (opposite to TBC)
	HVP	Horizontal and Vertical Position (same as CUP)
	IND	Index (similar to CUD)
	NEL	Next Line (similar to CNL)
	RI	Reverse Index (similar to CUU)
	TBC	Tab Clear
HP private sequences	VPA	Vertical Position Absolute
	VPR	Vertical Position Relative
	HPA	Horizontal Position Absolute (same as CHA)
	HPR	Horizontal Position Relative (same as CUF)
	HPHD	Home Down
	HPHU	Home Up
	HPSC	Save Cursor and attributes
	HPRC	Restore Cursor and attributes

These sequences are described below.

### CBT – Cursor Back Tabulation

$E_C [ P_n Z$

Moves the cursor horizontally backward along the active line to the  $P_n$ th preceding tab stop.  $P_n = 0$  or  $1$  moves the cursor to the first preceding tab position. The tab may be set programmatically using the HTS sequence or by the user using the ANSI Configuration Menu or the **margins/tabs/col** screen label. The cursor stops at column 1 if the  $P_n$ th tab stop is not found.

**CHA – Cursor Horizontal Absolute** $E_C [ P_n G$ 

Moves the active cursor position forward or backward along the active line to the specified column position.  $P_n = 0$  or  $1$  moves the cursor to the first position in the active line. A parameter value of  $P_n$  moves the cursor to the  $P_n$ th column of the active line. A value greater than the display capacity moves the cursor to the right edge of the display.

The CHA sequence has the same effect as the HPA (Horizontal Position Absolute) sequence.

**CNL – Cursor Next Line** $E_C [ P_n E$ 

Moves the active cursor position to the first column of the  $P_n$ th subsequent line.  $P_n = 0$  or  $1$  indicates the next line. A parameter value of  $P_n$  moves the cursor down by  $P_n$  lines. If line  $P_n$  is below the last displayed line, a roll up is performed (if permitted).

**CPL – Cursor Preceding Line** $E_C [ P_n F$ 

Moves the active cursor position to the first position of the  $P_n$ th previous line.  $P_n = 0$  or  $1$  indicates the previous line. A parameter value of  $P_n$  moves the cursor up by  $P_n$  lines. If line  $P_n$  is above the first line, a roll down is performed (if required).

**CPR – Cursor Position Report (terminal to host)** $E_C [ P_n ; P_n R$ 

The CPR sequence is generated in response to a DSR (Device Status Report) sequence requesting a cursor position report. The CPR sequence reports the active cursor position by means of the two parameters:

- 1) The first specifies the line.
- 2) The second specifies the column.

**CUB – Cursor Backward** $E_C [ P_n D$ 

Moves the cursor to the left by the specified number of columns.  $P_n = 0$  or  $1$  or no parameter, moves the cursor by one position. The cursor stops when column  $1$  is reached.

**CUD – Cursor Down** $E_C [ P_n B$ 

Moves the cursor down by the specified number of screen lines.  $P_n = 0$  or  $1$  or no parameter, moves the cursor down by one line. If an attempt is made to move the cursor past the bottom margin or line  $24$ , it will stop there (no rolling is performed).

**CUF – Cursor Forward** $E_C [ P_n C$ 

Moves the cursor to the right by the specified number of columns.  $P_n = 0$  or 1 or no parameter, moves the cursor right by one column. The cursor stops at the right edge of the screen.

**CUP – Cursor Position** $E_C [ P_n; P_n H$ 

Moves the cursor to the specified position. This sequence has two parameters:

- 1) The first specifies the line number.
- 2) The second specifies the column number.

The line numbering depends on the Multipage field in the ANSI Configuration Menu, and the set/reset state of the Origin Mode (HPOM):

- If the Multipage field is NO (one page), the cursor can only be positioned within the 24 lines of data displayed on the screen.
- If the HPOM (Origin Mode) is set, the cursor can only be positioned within the margins of the scrolling region defined using HPSTBM, and the lines are numbered with respect to the first line of the scrolling region.

If both parameters are 0 or no parameters are specified, the cursor moves to the home up position.

The CUP sequence has the same effect as the HVP (Horizontal and Vertical Position) sequence.

**CUU – Cursor Up** $E_C [ P_n A$ 

Moves the cursor up by the specified number of screen lines (without changing the column number).  $P_n = 0$  or 1 or no parameter, moves the cursor up by one line. If an attempt is made to move the cursor past the top margin, it will stop there (no rolling is performed).

**HTS – Horizontal Tab Set** $E_C H$ 

Sets one tab stop at the active cursor position.

If the user subsequently displays the ANSI Configuration Menu and presses **SAVE CONFIG**, this tab stop plus any others set in the menu will be stored in non-volatile memory.

These tab stops are only effective in ANSI/EM52 mode. They are ignored if the terminal is set to HP mode, but are re-instated if the terminal is subsequently re-set to ANSI or EM52 mode.

**HVP – Horizontal and Vertical Position** $E_C [ P_n; P_n f$ 

Moves the cursor to the specified position.

The HVP operates the same way as the CUP (Cursor Position) sequence.

**IND – Index**E<sub>C</sub> D

Moves the active cursor position down by one line (without changing the column number). If an attempt is made to move the cursor past the bottom margin or line 24, a roll up is performed (if required). If “Auto LF” is ON, then the active cursor position moves to column 1.

**NEL – Next Line**E<sub>C</sub> E

Moves the active cursor position to the first position on the next line (downward). If an attempt is made to move the cursor below the bottom margin or line 24, a roll up is performed (if required).

**RI – Reverse Index**E<sub>C</sub> M

Moves the active cursor position up by one line (without changing the column number). If an attempt is made to move the cursor above the top margin, a roll down is performed (if required).

**TBC – Tab Clear** $E_C [ P_s g$ 

Clears the tab position according to the parameter:

$P_s = 0$  or none    Clear tab at active position.  
 $P_s = 3$             Clear all horizontal tabs.

**VPA – Vertical Position Absolute** $E_C [ P_n d$ 

Moves the cursor to the specified line (without changing the horizontal position). A parameter of  $P_n$  moves the cursor vertically to line  $P_n$ . If there is no  $P_n$ th line, the cursor moves to the last available line.

**VPR – Vertical Position Relative** $E_C [ P_n e$ 

Moves the cursor downward by the specified number of lines (without changing the horizontal position).  $P_n = 0$ , or 1 or no parameter, moves the cursor one line down. If line  $P_n$  is below the last displayed line, a roll up is performed (if required).

**HPA – Horizontal Position Absolute** $E_C [ P_n \text{ `}$ 

Moves the active cursor position forward or backward along the active line to the specified column position.

**HPR – Horizontal Position Relative** $E_C [ P_n a$ 

Moves the cursor to the right by the specified number of columns.

**HPHD – Home Down** $E_C [ > 1 s$ 

Performs a Home Down, text is rolled up as required. This is equivalent to pressing keys [Shift] [↵].

**HPHU – Home Up** $E_C [ > 0 s$ 

Performs a cursor Home Up, text is rolled down as required. This is equivalent to pressing the [↵] key.

**HPSC – Save Cursor And Attributes** $E_C 7$ 

Causes the active cursor position, base/alternate character sets and display enhancement to be stored in temporary memory, where they are available for subsequent retrieval using the HPRC sequence. Note that the stored values are lost when the terminal is powered off or after a hard reset.

**HPRC – Restore Cursor And Attributes** $E_C 8$ 

Recalls the values stored by the last HPSC (Save Cursor) sequence. Note that no action is taken if an HPSC sequence has not been received since the terminal was last powered-up or reset.

## ANSI Display Control Sequences

The following display control sequences are responded to by the terminal in ANSI mode:

	Mnemonic	Description
ANSI specified sequences	NP	Next Page
	PP	Previous Page
	SD	Scroll Down
	SU	Scroll Up
HP private sequences	HPSTBM	Set top and bottom margin

### NP – Next Page

$E_C [ P_n U$

Causes the Pnth subsequent page of data in the display memory to be displayed. Pn = 0 or 1 or no parameter, displays the next page. A parameter value of Pn displays the Pnth subsequent page. The new page is displayed with the cursor in the home up position.

Note that the terminal ignores the NP sequence:

- 1) If the Multipage field in the ANSI Configuration Menu is NO (one page).
- 2) If a scrolling region (2 to 24 lines) is defined using HPSTBM.

### PP – Previous Page

$E_C [ P_n V$

Causes the Pnth preceding page of data in the display memory to be displayed. Pn = 0 or 1 or no parameter, displays the previous page. A parameter value of Pn displays the Pnth previous page. The new page is displayed with the cursor in the home up position.

The terminal ignores the PP sequence if the Multipage field in the ANSI Configuration Menu is NO (one page), or if a scrolling region is defined using HPSTBM.

**SD – Scroll Down** $E_C [ P_n T$ 

Causes all the data displayed on the screen to be moved down by the specified number of lines.  $P_n = 0$  or 1 or no parameter, scrolls the data down one line. A parameter value of  $P_n$  scrolls down  $P_n$  lines.

The cursor remains in the same position on the screen. As the bottom line of data is removed from the screen, another new line appears at the top. Scrolling stops when the first line on the screen is the first line in the display memory.

The HPSCLM (HP Scrolling Mode) sequence selects jump or smooth scrolling.

**SU – Scroll Up** $E_C [ P_n S$ 

Causes all the data displayed on the screen to be moved up by the specified number of lines.  $P_n = 0$  or 1 or no parameter, scrolls the data up one line. A parameter value of  $P_n$  scrolls up  $P_n$  lines.

The cursor remains in the same position on the screen. As the top line of data is removed from the screen, another new line appears at the bottom. If scrolling is continued until the bottom of display memory is reached, the last line in the memory is displayed at the top of the screen.

The HPSCLM (HP Scrolling Mode) sequence selects jump or smooth scrolling.



**HPSTBM – Set Top and Bottom Margins** $E_C [ P_n; P_n r$ 

At power-on, by default, the scrolling region is the entire screen. The HPSTBM sequence allows the scrolling region to be set to between 2 and 24 lines by means of two parameters:

- 1) The first specifies the line number of the first line of the scrolling region, and can have a value from 1 to 23.
- 2) The second specifies the line number of the bottom line of the scrolling region, and can have a value from 2 to 24.

The parameter values are screen lines, and are included in the scrolling region. The top/bottom margins are not saved in non-volatile memory.

The minimum size of the scrolling region is two lines (and the top margin must have a line number less than the bottom margin). The cursor is placed in the home position. (Refer to Origin Mode, HPOM.)

A parameter value of 0,0 sets the scrolling region equal to the entire screen.

**ANSI Editing Control Sequences**

The terminal responds to the following editing control sequences in ANSI mode:

	<b>Mnemonic</b>	<b>Description</b>
ANSI specified sequences	DCH	Delete Character
	DL	Delete Line
	ED	Erase in Display
	EL	Erase in Line
	IL	Insert Line

**DCH – Delete Character** $E_C [ P_n P$ 

Deletes the specified number of characters at the active cursor position.  $P_n = 0$  or 1 or no parameter deletes one character. Whenever a character is deleted, all characters to the right of the active cursor position (in the current line) are moved one character position left.

**DL – Delete Line** $E_C [ P_n M$ 

Deletes the specified number of lines at the active cursor position.  $P_n = 0$  or 1 or if no parameter, delete one line. Whenever a line is deleted, all the lines below the deleted line are moved one line up. The sequence is ignored when the cursor is outside the scrolling region.

**ED – Erase in Display** $E_C [ P_s J$ 

Erases some or all of the displayed characters according to the parameter:

$P_s = 0$  Erase from cursor to the end of the display (default).

$P_s = 1$  Erase from start of screen to cursor, inclusive.

$P_s = 2$  Erase all of the display.

The ED sequence does not move the cursor.

**EL – Erase in Line** $E_C [ P_s K$ 

Erases some or all the characters in the active line according to the parameter:

$P_s = 0$  Erase from cursor to end of line, inclusive (default).

$P_s = 1$  Erase from start of line to cursor, inclusive.

$P_s = 2$  Erase entire line.

The EL sequence does not move the cursor.

**IL – Insert Line** $E_C [ P_n L$ 

Inserts  $P_n$  blank lines at the active cursor position. All the subsequent lines are moved down one line. The sequence is ignored when the cursor is outside the scrolling region.

**ANSI Character Set Selection Sequences**

The following character set control sequences are responded to by the terminal in ANSI mode:

	Mnemonic	Description
ANSI specified sequences	SCS	Select Character Set
	SGR	Select Graphic Rendition
HP private sequences	HPDHL	Make this line double height, double width
	HPDWL	Make this line double-width
	HPSSL	Make this line standard size

**SCS – Select Character Set**

$E_C$  (  $P_s$  – select base character set (G0)

$E_C$  )  $P_s$  – select alternate character set (G1)

Where  $P_s$  can have the values:

$P_s = B$  = configured language.

$P_s = 0$  = ANSI line drawing set.

$P_s = 3$  = HP line drawing set.

At power-on or after a hard reset, the default base character set (G0) is always the configured language set. The default alternate character set (G1) is always the ANSI line drawing set.

To invoke the base set (previously selected by the appropriate ANSI escape sequence) while operating in the alternate set, issue a Shift-in ( $S_i$ ) control character.

The alternate set (G1) is enabled by the Shift-out ( $S_o$ ) control character. Unlike HP mode, the alternate set stays enabled until you perform a hard reset or send a Shift-in ( $S_i$ ) control character.

**SGR – Select Graphic Rendition**

$E_C$  [  $P_s$ ; . . . ;  $P_s m$

Select video attribute(s) according to parameter value(s):

$P_s = 0$  Turn off all attributes.

$P_s = 1$  Half-Bright.

$P_s = 4$  Underline.

$P_s = 5$  Blinking.

$P_s = 7$  Inverse Video.

Attributes may be mixed in the same sequence. The attributes remain in effect until another SGR sequence is encountered.

**HPDHL – Double Size Line**

$E_C \# P_s$

Sets the line containing the cursor to be one half of a double height double width line according to the parameter:

$P_s = 3$  sets the line containing the cursor to become the top half of a double height double width line.

$P_s = 4$  sets the line containing the cursor to become the bottom half of a double height double width line.

A double height double width line can contain up to 40 double-size characters.

The sequences must be used as a pair on adjacent lines.

The same character output (string) must be sent to both lines to form full double-height characters. Since there can be only 40 double-size characters per line.

For example, to set characters “THAN” as double size characters, use the sequences:

EC # 3	THAN	CR LF	→	THAN	top half of line
EC # 4	THAN	CR LF	→		bottom half of line

### HPDWL – Double-Width Line

EC # 6

Causes the line containing the cursor to become a double width single height line. A double-width line can contain up to 40 normal-size characters. If the line was single width, all characters to the right of the center of the screen (beyond the 40th) are not displayed but are retained in memory.

### HPSSL – Single Size Line

EC # 5

Causes the line containing the cursor to become a single width, single height line. If the line was one half of a double-height line, the other half of the double-height line is unaffected (and the corresponding half of the double-size character will remain on the display).

## ANSI Terminal Status Sequences

In ANSI mode, the following status control sequences are responded to or sent by the terminal:

ANSI sequences	{ DA	Device Attributes
	DSR	Device Status Request/Report
HP private sequences	{ HPID	Identity (same as DA)
	HPREQTPARM	Request Parameters
	HPREPTPARM	Report Parameters (sent in response to HPREQTPARM)

### DA – Device Attributes (host to terminal)

EC [ 0 c  
or EC [ c

The host requests the terminal to send an identification sequence. The terminal will respond with:

EC [ ? 1 ; 2 c

**DSR – Device Status Report (host to terminal and terminal to host)** $E_C [ P_s n$ 

Used in both host-to-terminal and terminal-to-host transfers to report the terminal status by means of the parameters. This sequence has four parameters:

- CPU to terminal:  $P_s = 5$  Return status report using DSR sequence.  
 $P_s = 6$  Return cursor position report using a CPR sequence.
- Terminal-to-host:  $P_s = 0$  Terminal ready, no malfunctions.  
(response to DSR  $P_s = 3$  Malfunction detected, retry.  
from host)

**HPID – Identify** $E_C Z$ 

The HPID sequence has the same effect as the DA (Device Attributes) sequence.

**HPREQTPARM – Request Terminal Parameters (host to terminal)** $E_C [ P_s x$ 

Sent by the host to request the terminal's operating parameters; where  $P_s$  can have the value 0 or 1. Parameters are reported using a HPRPTPARM sequence.

Table 10.5 HPRPTPARM Report Parameters Mnemonics

Parameter Mnemonic	Parameter Value	Meaning
< par >	1	No parity set
	2	Space parity
	3	Mark parity
	4	Parity set to ODD
	5	Parity set to EVEN
< nbits >	1	8 bits per character
	2	7 bits per character
< xspeed > (terminal transmit speed) and < rspeed > (terminal receive speed)	16	110 Baud
	24	134.5 Baud
	32	150 Baud
	48	300 Baud
	56	600 Baud
	64	1200 Baud
	72	1800 Baud
	88	2400 Baud
	72	4800 Baud
	112	9600 Baud
	120	19200 Baud

**HPRPTPARM Report Terminal Parameters (terminal to host)**

$$E_C [ 3; <par>; <nbits>; <xspeed>; <rspeed>; 1; 0 x$$

Returned after a HPREQPARM sequence had been received to Report the terminal parameters. The sequence parameter mnemonics are listed in Table 10-5. Note that on the 2392A,  $<xspeed>$  and  $<rspeed>$  always have the same value.

**ANSI Terminal Control Sequences**

ANSI specified	{ MC RIS	Media Copy Reset to initial state
HP private sequences	{ HPKPAM HPKPNM HPLF	Set Keypad to Application Mode Set Keypad to Numeric Mode Load flags

**MC – Media Copy**

$$E_C [ P_s i$$

Controls the transfers between the terminal and the external device (provided the terminal is fitted with a second port) according to the parameter  $p_s$ :

- $P_s = 0$  Copy All: Copy all data in display memory to printer, from line containing the cursor to end of memory.
- $P_s = 5$  Turn on Log Bottom Mode: Incoming data is displayed on the screen, and when a line feed is encountered (either received from the host or as the result of a Wraparound), the line of data is sent to the printer.
- $P_s = 4$  Turn Off Log Bottom Mode: Incoming data is no longer sent to the printer as it is received.

**RIS – Reset to Initial State**

$$E_C c$$

Terminal performs a hard reset (equivalent to power-on). The terminal is then set as follows:

- 1) The cursor is displayed at the top left-hand corner of the screen (column 1, line 1), and the display memory is cleared.
- 2) The eight labels associated with the MODES function are displayed at the bottom of the screen.
  - a) Four operating controls associated with the MODES labels are inactive (no asterisk in label): LINE MODIFY, SMOOTH SCROLLING, MEMORY LOCK and DISPLAY FUNCTIONS.
  - b) The remaining three MODES operating controls (MODIFY ALL, REMOTE MODE and AUTO LF), plus the parameters set using the terminal's configuration menu, are set as you configured them before powering OFF (these parameters are stored in non-volatile memory).

3) Certain keys and parameters are reset as follows:

- a) The keyboard is unlocked.
- b) The [Caps] key is reset (set to lower-case). The terminal may still generate upper-case characters if the Terminal Configuration Menu's CAPS LOCK feature is ON.
- c) The User Keys retain the definitions given to them before powering off.
- d) Cursor key and numeric pad application modes are reset.
- e) The scrolling region defaults to the entire screen.
- f) Tab stops, backspace and multipage parameter as specified in the ANSI Configuration Menu.
- g) The "insert character" edit function is switched OFF.
- h) The Record Mode is switched OFF (this stops the terminal from automatically sending data to an external device).

### HPKPAM – Keypad Application Mode

$E_C =$

Sets the numeric keypad to Application Mode, see Table 10-3.

### HPKPNM – Keypad Numeric Mode

$E_C >$

Sets the numeric keypad to Numeric Mode, see Table 10-3.

### HPLF – Load Flags (HP Private)

$E_C [Ps q]$

Controls the four programmable flags in the display Status Line as follows:

Ps Value	Effect on flags
0	Clear all flags
1	Illuminate flag "L1"
2	Illuminate flag "L2"
3	Illuminate flag "L3"
4	Illuminate flag "L4"

## ANSI Terminal Mode Selection Sequences

During ANSI operations, various modes (or terminal controls) are available that operate in a similar manner to the straps in HP mode. These ANSI modes are listed in Table 10-6. The modes labeled "set/reset" (in the value column) can be set and reset using the SM (Set Mode) and RM (Reset Mode) sequences, respectively. The individual modes are described below.

Table 10-6 also lists other modes defined by ANSI Standard X3.64-1974. These modes, however, have a pre-defined fixed state that cannot be altered from the host computer or by the user. These modes are not affected by the SM and RM sequences.

Table 10-6. ANSI and HP Private Modes

Mnemonic	Parameter (Ps)	Value	Function	Power-on Default Value
ANSI-specified Modes				
GATM	1	N/A	Guarded Area Transfer Mode	Reset Reset Reset
KAM	2	Set/Reset	Keyboard Action Mode	
CRM	3	Set/Reset	Control Representation Mode	
IRM	4	Set/Reset	Insert/Replace Mode	
SRTM	5	Reset	Status Report Tranfer Mode	
ERM	6	Set	Erasure Mode	
VEM	7	N/A	Vertical Editing Mode	
HEM	10	N/A	Horizontal Editing Mode	
PUM	11	Reset	Positioning Unit Mode	
SRM	12	Set	Send-Receive Mode	
FEAM	13	Reset	Format Effector Action Mode	
FETM	14	Reset	Format Effector Transfer Mode	
MATM	15	N/A	Multiple Area Transfer Mode	
TTM	16	N/A	Transfer Termination Mode	
SATM	17	N/A	Selected Area Transfer Mode	
TSM	18	Reset	Tabulation Stop Mode	
EBM	19	Reset	Editing Boundary Mode	Reset
LNM	20	Set/Reset	Line Feed/New Line Mode	
HP Private Modes				
HPCKM	? 1	Set/Reset	Cursor Keys Mode	Reset
HPANM	? 2	Set/Reset	ANSI Mode	Reset
HPSCLM	? 4	Set/Reset	Scrolling Mode	Reset
HPOM	? 6	Set/Reset	Origin Mode	Set
HPAWM	? 7	Set/Reset	Autowrap Mode	Reset
HPMPM	> 1	Set/Reset	Multipage Mode	Set
HPMLM	> 2	Set/Reset	Memory Lock Mode	Reset



**RM – Reset Mode** $E_C [ P_s; \dots; P_s ]$ 

Resets the terminal modes defined by the parameters  $P_s$ . The available parameters comprise all of the modes labeled “set/reset” in the “value” column in Table 10-6 (these include ANSI and HP private modes). RM is the opposite of the SM (Set Mode) sequence. These modes are described in detail below.

**SM – Set Mode** $E_C [ P_s; \dots; P_s ]$ 

Sets the terminal modes defined by the parameter  $P_s$ . The available parameters comprise all of the modes labeled “set/reset” in the “value” column in Table 10-6 (these include ANSI and HP private modes). SM is the opposite of the RM (Reset Mode) sequence. These modes are described in detail below.

**KAM – Keyboard Action Mode** $E_C [ 2 \text{ h (Set)} ]$  $E_C [ 2 \text{ l (Reset) (Default state)} ]$ 

The set state disables the keyboard, and displays the message “Kb Lockd” in the screen Status Line.

In the reset state, the keyboard is enabled.

**CRM – Control Representation Mode** $E_C [ 3 \text{ h (Set)} ]$  $E_C [ 3 \text{ l (Reset) (Default state)} ]$ 

In the set state, control characters are displayed symbolically and the terminal does not execute the control function. A CR character, however, displays a  $\mathcal{C}\mathcal{R}$  symbol and also causes a line feed and carriage return to be performed.

In the reset state, control characters are executed and not displayed.

The Control Representation Mode can also be set/reset using the terminal’s **DISPLAY FUNCTION** screen-labeled function key.

**IRM – Insert/Replace Mode** $E_C$  [ 4 h (Set) $E_C$  [ 4 l (Reset) (Default state)

Controls how entered or received characters affect characters previously displayed at the active cursor position.

In the reset state, each new displayable character overwrites the character previously displayed at the active cursor position. The cursor is then moved to the right (if it is not at the right margin), and no other character will be affected.

In the set state, each new character causes all characters on that line at or to the right of the active cursor position to be shifted one place to the right. The new character is inserted at the active position, and the cursor is then moved to the right. The screen status line displays the message “Inschar” when this function is active.

The IRM mode can also be set/reset using the terminal’s [Insert char] key.

**LNM – New Line Mode** $E_C$  [ 20 h (Set) $E_C$  [ 20 l (Reset) (Default state)

Controls the character(s) generated by the [Return] key.

In the set state, the characters “C<sub>R</sub>” and “L<sub>F</sub>” are generated.

In the reset state, only the character “C<sub>R</sub>” is generated.

The LNM mode can also be set using the AUTO LF screen label.

**HPCKM – HP Cursor Keys Mode** $E_C$  [ ? 1 h (Set) $E_C$  [ ? 1 l (Reset) (Default state)

In the reset state, the cursor keys send ANSI cursor movement control codes.

In the set state, the cursor keys send application function control codes; see Table 10-2.

**HPANM – HP/ANSI/EM52 Mode**

$\text{E}_C [ ? 2 1$  Leave ANSI mode and enter EM52 mode

$\text{E}_C <$  Leave EM52 mode and enter ANSI mode

The terminal has the capability of operating with EM52 control sequences. When in ANSI mode, “EM52 mode” is entered with a standard ANSI control sequence ( $\text{E}_C [ ? 2 1$ ); when in the EM52 mode, ANSI operation can be resumed with  $\text{E}_C <$ .

The other mode-changing sequences which will switch the terminal between HP Mode and ANSI Mode are:

$\text{E}_C \& k 1 \setminus$  Leave HP mode and enter ANSI mode

$\text{E}_C \& k 0 \setminus$  Leave ANSI mode and enter HP mode

Note: Entering ANSI Mode from HP Mode causes the following to happen: Block Mode is disabled, Memory Lock is turned off, the display is cleared, and the cursor is homed up.

**HPSCLM – HP Scrolling Mode**

$\text{E}_C ( ? 4 h$  (Set)

$\text{E}_C [ ? 4 l$  (Reset) (Default state)

The reset state causes lines to jump when scrolling is used.

The set state causes lines to flow “smoothly” when scrolling is used.

The HPSCLM mode can also be set/reset using the **SMOOTH SCROLL** screen-labeled function key.

**HPOM – HP Origin Mode**

$\text{E}_C [ ? 6 h$  (Set)

$\text{E}_C [ ? 6 l$  (Reset) (Default state)

In the reset state, the origin is the upper-left character position of the screen. Line and column numbers are therefore independent of top and bottom margins set using the HPSTBM sequence. The cursor may be positioned outside the margins with a Cursor Position (CUP) or Horizontal and Vertical Position (HVP) control sequence.

The set state causes the origin to be at the upper-left character position within the top and bottom margins set using the HPSTBM. The cursor may not be positioned outside the margins set by the HPSTBM sequence.

**HPAWM – HP AutoWrap Mode** $E_C [ ? 7 h$  (Set) (Default state) $E_C [ ? 7 l$  (Reset)

In the reset state, automatic wraparound is not performed and when the cursor reaches the right margin it stays there (until explicitly moved). Consequently, characters received when the cursor is at the right margin overwrite any existing character.

In the set state, automatic wraparound is performed. Characters overflowing the right margin are written at the start of the next line, a scroll up is performed if required and if permitted.

**HPMPM – HP Multipage Mode** $E_C [ > 1 h$  (Set) (Default state) $E_C [ > 1 l$  (Reset)

The set state selects multiple paging.

The reset state causes the terminal to become a single page terminal, with 24 lines of display memory.

When the terminal is either set or reset in this mode, the display is cleared and the cursor is homed up.

The HPMPM sequence temporarily overrides the Multipage parameter set in the ANSI Configuration Menu. The terminal returns to the menu setting after power-on or a hard reset.

**HPMLM – HP Memory Lock Mode** $E_C [ > 2 h$  (Set) $E_C [ > 2 l$  (Reset)

The set state turns on memory lock, which then functions similar to the way it does in HP mode.

The reset state turns Memory Lock off.

Memory Lock Mode can also be reset using the terminal's **MEMORY LOCK** screen label.

## EM52 CONTROL SEQUENCE SUMMARY

The following sub-sections describe the control sequences recognized and processed by the terminal when in EM52 mode.

Note that EM52 mode can only be entered from ANSI mode, using the ANSI HPANM control sequence (it is not possible to enter the EM52 mode directly from HP mode).

### Cursor Up

$E_C A$

Moves the active cursor position upward by one line (without changing the column number). If an attempt is made to move the cursor past the top margin, it stops there (at the top margin).

### Cursor Down

$E_C B$

Moves the active cursor position down by one line (without changing the column number). If an attempt is made to move the cursor past the bottom margin, it stops there (at the bottom margin).

### Cursor Right

$E_C C$

Moves the active cursor position one column to the right. If an attempt is made to move the cursor past the right margin, it stops at the right margin.

### Cursor Left

$E_C D$

Moves the active cursor position one column to the left. If an attempt is made to move the cursor past the left margin, it stops at the margin.

### Cursor to Home

$E_C H$

Moves the cursor to the home-up position.

## Reverse Line Feed

$E_C I$

Moves the active cursor position upward by one line (without changing the column number). If an attempt is made to move the cursor past the top margin, a roll down is performed (if required).

## Erase to End Of Screen

$E_C J$

Erases all characters from the active cursor position to the end of the screen. The active cursor position is not changed.

## Erase to End Of Line

$E_C K$

Erases all characters from the active cursor position to the end of the line. The active cursor position is not changed.

## Direct Cursor Address

$E_C Y P_n P_n$

Move the cursor to the specified position. This sequence has two parameters:

- 1) The first specifies the line number.
- 2) The second specifies the column number.

When specifying a line or column number, the parameter used must be sent as ASCII code of the value of the line/column number plus octal 37 (hexadecimal 1F). For example: to specify line or column number "1", a space should be used (hex. 1F + 1); to specify line or column "2" a "!" should be used (hex. 1F + 2).

## Identify

$E_C Z$

Causes the terminal to return its identifier sequence to the host CPU. The returned identifier sequence is:  $E_C / Z$ .

**Enter Alternate Keypad Mode** $E_C =$ 

Sets the numeric keypad to application mode. (Refer to table 10-3.)

**Exit Alternate Keypad Mode** $E_C >$ 

Sets the numeric keypad to numeric mode. (Refer to table 10-3.)

**Enter ANSI Mode** $E_C <$ 

Leave EM52 mode and enter ANSI mode.

(

(

(



# Escape Codes

## A

KEY(S)	CODE	FUNCTION
TERMINAL CONTROL FUNCTION		
[Print Enter] (as used in Local mode)	$\text{E}_\text{C}$ 0	Copy memory to destinations(s)
[User System] , margins / , SET tabs / col TAB	$\text{E}_\text{C}$ 1	Set tab
[User System] , margins / , CLEAR tabs / col TAB	$\text{E}_\text{C}$ 2	Clear tab
[User System] , margins / , CLR ALL tabs / col TABS	$\text{E}_\text{C}$ 3	Clear all tabs
[User System] , margins / , LEFT tabs / col MARGIN	$\text{E}_\text{C}$ 4	Set left margin
[User System] , margins / , RIGHT tabs / col MARGIN	$\text{E}_\text{C}$ 5	Set right margin
	$\dagger \text{E}_\text{C}$ 6	Define alphabetic only field
	$\dagger \text{E}_\text{C}$ 7	Define numeric-only field
	$\dagger \text{E}_\text{C}$ 8	Define unrestricted (all characters) field
[User System] , margins / , CLR ALL tabs / col MARGINS	$\text{E}_\text{C}$ 9	Clear all margins
	$\text{E}_\text{C}$ @	Delay one second
[▲]	$\text{E}_\text{C}$ A	Cursor up
[▼]	$\text{E}_\text{C}$ B	Cursor down
[►]	$\text{E}_\text{C}$ C	Cursor right
[◄]	$\text{E}_\text{C}$ D	Cursor left
[Shift] [CTRL] [Reset]	$\text{E}_\text{C}$ E	Hard reset
[Shift] [▽]	$\text{E}_\text{C}$ F	Cursor home down
	$\text{E}_\text{C}$ G	Move cursor to left margin
[▽]	$\text{E}_\text{C}$ H	Cursor home up
[Tab ►]	$\text{E}_\text{C}$ I	Horizontal tab (tab right)
[Clear display]	$\text{E}_\text{C}$ J	Clear display from cursor to end of memory

$\dagger$  Valid only on the HP 2394A terminal

Continued . . .

KEY(S)	CODE	FUNCTION
TERMINAL CONTROL FUNCTION (Continued)		
[Clear line]	$E_C$ K	Clear line from cursor to end of line
[Insert line]	$E_C$ L	Insert line
[Delete line]	$E_C$ M	Delete line
[Delete char]	$E_C$ P	Delete character
[Insert char]	$E_C$ Q	Start insert character mode
[Insert char]	$E_C$ R	End insert character
[Shift] [▲]	$E_C$ S	Roll up
[Shift] [▼]	$E_C$ T	Roll down
[Next]	$E_C$ U	Next page
[Prev]	$E_C$ V	Previous page
	$E_C$ W	Format mode on
	$E_C$ X	Format mode off
[User System], Modes , DISPLAY FUNCTION*	$E_C$ Y	Display Functions mode on
[User System], Modes , DISPLAY FUNCTNS	$E_C$ Z	Display Functions mode off
	$E_C$ [	Start unprotected field
	$E_C$ ]	End unprotected/transmit- only fields
	$\dagger E_C$ {	Start transmit-only field
	$E_C$ ^	Primary terminal status request
	$E_C$ *s^	Terminal I.D. status request
	$E_C$ `	Sense cursor position (relative)
	$E_C$ a	Sense cursor position (absolute)
	$E_C$ b	Unlock keyboard
	$E_C$ c	Lock keyboard
	$E_C$ d	Transmit a block of text to computer
$\dagger$ Valid only on the HP 2394A terminal.		

Continued . . .

KEY(S)	CODE	FUNCTION
<b>TERMINAL CONTROL FUNCTION (Continued)</b>		
	$E_C$ f	Modem disconnect
[Shift] [Reset Break]	$E_C$ g	Soft reset
[▽]	$E_C$ h	Cursor home up
[Shift] [Tab ◀▶]	$E_C$ i	Backtab
[CTRL] [Menu]	$E_C$ j	Display User Key Menu and begin User Key Definition Mode
[User System] or [Shift] [User System]	$E_C$ k	Restore normal display and end User Key Definition Mode
Modes , MEMORY LOCK*	$E_C$ l	Begin Memory Lock mode
Modes , MEMORY LOCK	$E_C$ m	End Memory Lock mode
[f1]	$E_C$ p	Default definition for user definable function key f1
[f2]	$E_C$ q	Default definition for user definable function key f2
[f3]	$E_C$ r	Default definition for user definable function key f3
[f4]	$E_C$ s	Default definition for user definable function key f4
[f5]	$E_C$ t	Default definition for user definable function key f5
[f6]	$E_C$ u	Default definition for user definable function key f6
[f7]	$E_C$ v	Default definition for user definable function key f7
[f8]	$E_C$ w	Default definition for user definable function key f8
[User System] , service , TERMINAL keys , TEST	$E_C$ z	Initiate terminal self test
	$E_C$ ~	Secondary terminal status request
	$E_C$ &x1C	Enable the send cursor position mode
	$E_C$ &x0C	Disable the send cursor position mode

### CURSOR CONTROL OPERATIONS

#### NOTE

Columns and rows are numbered starting with 0 as the leftmost column and the top row.

$E_C$  &a<col> c <row> Y

Moves the cursor to column "col" and screen row "row" on the screen (screen relative addressing.)

$E_C$  &a<col> c <row> R

Moves the cursor to column "col" and row "row" in memory (absolute addressing).

$E_C$  &a±<col> c±<row> Y

Moves the cursor by columns "col" and rows "row" (on the screen) relative to its present position ("col" and "row" are signed integers). A positive number indicates right or downward movement and a negative number indicates left or upward movement.

$E_C$  &a±<col> c±<row> R

Moves the cursor by columns "col" and rows "row" relative to its present cursor position in memory ("col" and "row" are signed integers). A positive number indicates right or downward movement and a negative number indicates left or upward movement.

†  $E_C$  &a<Col 1> s <Col 2> s...<Col n> S

Vertical tab regions only: When in format mode, the normal tab sequence may be altered by splitting display memory into two or more vertical regions up to maximum of 80. <Col n> is the starting column number of the n'th + 1 vertical region. Column numbers are specified relative to zero.

†  $E_C$  &a<Col 1> s <Col 2> s...<Col n> s <Row 1> I

Top horizontal tab range: Vertical tab regions may be defined for a range of rows beginning at the top of display memory and ending on a row number specified by <row 1>. Row numbers are specified relative to zero.

†  $E_C$  &a<Row 1> i <Col 1> s <Col 2> s...<Col n> S

Bottom horizontal tab range: Vertical tab regions may be defined for a range of rows beginning on a specified row <Row 1> and ending at the end of display memory.

†  $E_C$  &a<Row 1> i <Col 1> s <Col 2> s...<Col n> s <Row 2> I

Horizontal mid-range tabs: Vertical tab regions may be defined for a range of rows beginning on a specified row, <Row 1>, and ending on a specified row, (the row directly before <Row 2>).

CONFIGURATION OPERATIONS			
$E_C$ &q 0L	Unlock all configuration menus.	} Sequence retained in non-volatile memory.	
$E_C$ &q 1L	Lock all configuration menus; plus functions <b>MODIFY ALL</b> , <b>Block MODE</b> , <b>REMOTE MODE</b> and <b>AUTO LF</b> .		
ESCAPE SEQUENCE	DESCRIPTION	VALUE	<x>
$E_C$ &k <x> A	AUTO LF	OFF ON	x = 0 x = 1
$E_C$ &k <x> B	BLOCK MODE	OFF ON	x = 0 x = 1
$E_C$ &k <x> C	Caps Lock	OFF ON	x = 0 x = 1
$E_C$ &k <x> D	Bell	OFF ON	x = 0 x = 1
$E_C$ &k <x> I	None/8 bits	NO (7 bits + existing parity) YES (none/8 bits)	x = 0 x = 1
$E_C$ &k <x> K	Auto Keyboard Lock	OFF ON	x = 0 x = 1
$E_C$ &k <x> L	Local Echo	OFF ON	x = 0 x = 1
$E_C$ &k <x> M	MODIFY ALL	OFF ON	x = 0 x = 1
$E_C$ &k <x> N	SPOW Latch	OFF ON	x = 0 x = 1
$E_C$ &k <x> P	Caps Mode ([Caps] key)	OFF ON	x = 0 x = 1
$E_C$ &k <x> R	REMOTE	OFF ON	x = 0 x = 1
$E_C$ &k <x> [	SMOOTH SCROLL	OFF ON	x = 0 x ≠ 0
$E_C$ &k <x> ]	[Select] key	OFF ON	x = 0 x = 1
$E_C$ &k <x> \	TermMode	HP ANSI	x = 0 x = 1
$E_C$ &s <x> A	XmitFunctn (A)	NO YES	x = 0 x = 1
$E_C$ &s <x> B	SPOW (B) (enable SPOW latch)	NO YES	x = 0 x = 1
$E_C$ &s <x> C	InhEolWrp (C)	NO YES	x = 0 x = 1
$E_C$ &s <x> D	Line/Page (D)	LINE PAGE	x = 0 x = 1
$E_C$ &s <x> G	InhHndShk (G)	NO YES	x = 0 x = 1
$E_C$ &s <x> H	Inh DC2 (H)	NO YES	x = 0 x = 1
$E_C$ &s <x> N	Esc Xfer (N)	NO YES	x = 0 x = 1
$E_C$ &s <x> X	Data Speed Select SR(CH)	LO HI	x = 0 x = 1
$E_C$ &s <x> Z	Check Parity	NO YES	x = 0 x = 1

† Valid only on the HP 2394A terminal.

## DATA TRANSFER OPERATIONS

The following escape sequences control data transfer to and from the external printers and display memory. Only effective if printer port present. The HP 2392A/94A does NOT support an internal printer.

**E<sub>C</sub> &p|<x>d|<x>D**      Selects device "<x>" as the destination device (displays "\*" in selected "to" device screen label and turns off previous "to" device).  
x=3 display, x=4 external device and [<x>d] = optional.

For the following sequences, if "<x> d" (destination) parameter is not specified, the selected "to" devices in the screen labels are assumed to be the destinations.

**E<sub>C</sub> &p <a>d <b>d <Y>**      Copies "<Y>" amount of data to destination devices "<a>" and "<b>". As many destinations as desired can be specified.

**<a> and <b>                      DEVICE**

- 3    Display.
- 4    External printer.

**<y>                                  ACTION**

- B    Copy the line in which the cursor is located.
- F    Copy the display screen from the line in which the cursor is located (cursor line) to the last displayed line.
- M    Copy the contents of display memory from the cursor line to the end of memory.

**NOTE:** In format mode, only the unprotected fields are printed. The 2394A will optionally print all of display memory when print <all> is selected in the terminal configuration menu.

**E<sub>C</sub> &p <x> ^**      Requests the status of device "<x>".

**<x>                                  DEVICE**

- 4    External printer.

**E<sub>C</sub> &p <x>p <y>u <z>C**      Performs the action specified by "<z>" on device "<y>".

**<z>                                  ACTION**

- 0    Generate 1 form feed.
- 1    Space number of lines specified by "<x>".
- 2-10 Generates 1 form feed.
- 11   Turn on Log Bottom mode.
- 12   Turn on Log Top mode.
- 13   Turn off any logging mode.
- 14   (ignored).
- 15   (ignored).
- 16   (ignored).
- 17   (ignored).
- 18   (ignored).
- 19   (ignored).
- 20   Turn on Record Mode.

DATA TRANSFER OPERATIONS (Continued)	
<b>&lt;y&gt;</b>	<b>DEVICE</b>
4	External printer.
3	Display.
<b>&lt;x&gt;p</b>	Optional parameter which can define the character to turn off Record mode. Alternatively, it may be used in conjunction with 1C (parameter <z>) to specify the number of lines to advance e.g. to generate a ten line space on the external device use $\text{E}_\text{C}\&\text{p } 10\text{p } 4\text{u } 1\text{C}$ .
$\text{E}_\text{C}\&\text{p } <\text{a}>\text{d } <\text{b}>\text{d } <\text{x}>\text{W } <\text{data string}>$	Transfers "x" bytes of the data string from the computer to the selected destination device in binary form.
$\text{E}_\text{C}\&\text{p } <\text{a}>\text{d } <\text{b}>\text{d } \text{W } <\text{data string}>$	Transfers the data string, in ASCII form, from the computer to the printer selected as the destination device. The string is terminated either by the 256th byte or by an ASCII line feed character.

† Forms Cache (Storage)	
$E_C$ & q 4te2{<x>L	Selects the size of forms cache (storage). Where “x” is the number of 256 byte blocks. It is a decimal value between 0 and 95. The maximum value is a function of the amount of display memory installed and the amount of memory allocated for data comm buffers.
$E_C$ & p9^<form#>p	Returns the forms cache status condition.
$E_C$ & p9u^<form#>p <x>	General form to define, purge, and transfer a form. Where “9u” is the device assignment, “form#” specifies the form number from 1-255, and “x” is the following:
<x>	MEANING
0L or L	purge specified form
F	transfer a form to the screen
<form-size>L<form-contents>	define a form (known length)
<<form-contents>>L	define a form (unknown length, multi-blocked)
<b>EXAMPLES:</b>	
To define a form, use either of the two escape sequences.	
$E_C$ & p9u<form#>p<form-size>L<form-contents>	
or	
$E_C$ & p9u<form#>p<<form-contents>>L	
To purge a form, use the following escape sequence:	
$E_C$ & p9u<form#>p0L	
To transfer a form to the screen, use the following escape sequence:	
$E_C$ & p9u<form#>pF	
An “S” status will be returned at the completion of the transfer sequence, unless an error is detected, another forms activity is in progress, or if the forms cache memory is full. In these cases, an “F” status will be returned.	

† Valid only on the HP 2394A terminal.



FORMAT MODE	
$E_c[$	Starts a field.
$E_c]$	Ends the field.
$\dagger <E_c \& e <x>e <y>$	Starts an edit rule of type "y" with valid character type "x" specified.
<b>y MEANING</b>	
<b>R</b> Required	
<b>T</b> Total fill	
<b>J</b> Justify	
<b>P</b> Modify data tag type	
<b>x TYPE</b>	
0 All characters (default)	
1 Alphabetic-only	
2 Auto upshift	
3 Alphanumeric	
4 Integer-only	
5 Signed decimal	
6 Implied decimal	
7 Constant	
8 Integer with fill	
9 Signed decimal with fill	
10 Implied decimal with fill	
11 Numeric-only	
$\dagger E_c \& k <n>X$	Selects the decimal type. For "n" = 0, the decimal type is US (default); for "n" = 1, the decimal type is European.
$\dagger E_c \& k <x>Y$	Selects the number of implied decimal digits "x", where "x" can be any digit from 0 to 9 and the default is 2.
$\dagger E_c \& k <x>Z$	The data transmitted when the ENTER key is pressed is selected by an "x"

$\dagger$  Valid only on the HP 2394A terminal.

### FUNCTION KEY AND ERROR MESSAGE OPERATIONS

To enable and disable the function keys ([f1] thru [f8]), use the following escape sequence:

$\text{F}_C \& j < x >$

$< x >$

#### MEANING

- A Display the modes set of function key labels.
- B Enable the user function keys. (The user key labels are displayed).
- C Clear message and restore the current key labels.
- @ Remove the function key labels from the screen. The user function keys, however, are still active.

To enable or disable the Function Control keys:

S Disable the [User System] and [Menu] keys, and label "modes".

R Enable the [User System] and [Menu] keys, and label "modes".

To define the function keys:

$\text{F}_C \& f < \text{attribute} > a < \text{key} > k < \text{label length} > d < \text{string length} > L < \text{label} > < \text{string} >$

(Sequence retained in non-volatile memory).

TERM	SYMBOL	MEANING	DEFAULT
$< \text{attribute} > (a)$	0	Normal (N)	0
	1	Local only (L)	
	2	Transmit only (T)	
$< \text{key} > (k)$	1	[f1] function key	1
	2	[f2] function key	
	3	[f3] function key	
	4	[f4] function key	
	5	[f5] function key	
	6	[f6] function key	
	7	[f7] function key	
	8	[f8] function key	

Continued ...

FUNCTION KEY AND ERROR MESSAGE OPERATIONS (Continued)			
TERM	SYMBOL	MEANING	DEFAULT
<label length> (d)	0 thru 16	Number of characters in the label. (32 diacritical combinations)	0
<string length> (L)	0 thru 80	Number of characters in the string. (160 diacritical combinations)	1
	- 1	Clears the content of the string.	
‡ <label>	(none)	The label is entered at this point in the sequence.	
‡ <string>	(none)	The character string is entered at this point in the sequence.	
$E_C$ & f <key> E		Initiate execution of key: - 1 = [Enter], 1 = [f1] to 8 = [f8] ‡ 0 = [Return]	
To replace the function key definition with your own message:			
$E_C$ & j <string length> L <message>			
<string length> - A number indicating the number of characters in the string (up to 160 characters, 320 diacritical combinations).			
<message> - The content of the message			

‡ The <label> and <string> may contain display enhancements on the HP 2394A terminal.

† Valid only on the HP 2394A terminal.

### DISPLAY ENHANCEMENTS OPERATIONS

To start and end display enhancements

$E_C \& d \< \text{char} \>$  Selects the display enhancement indicated by “char” to begin at the present cursor position.

		“< char>”															
		@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Half-Bright										X	X	X	X	X	X	X	X
Underline						X	X	X	X					X	X	X	X
Inverse Video				X	X			X	X			X	X			X	X
Blinking			X		X		X		X		X		X		X		X
End Enhancement	X																

$E_C \& d S$  Start Security field.

$E_C \& d s \< \text{char} \>$  Enable Security field and other selected enhancement(s).

#### Screen Control

$E_C \& w 12F$  Turns on the display screen (top 24 lines).

$E_C \& w 13F$  Turns off the display screen (top 24 lines).

## CONTROL CHARACTERS

Control characters (hexadecimal 00 to 1F inclusive) are excluded from the control sequence syntax, but may be embedded within a block of text. The following table lists the control characters processed by the terminal, all other control characters are ignored.

Control Character	Hex Code	Terminal Action
ENQ	05	Transmits an ACK character if the ‘‘EnqAck’’ field in the datacomm configuration menu is set to YES.
BEL	07	Sounds the bell.
$B_S$	08	Moves the cursor one position to the left, stopping at the left margin of current line.
$H_T$	09	Moves the cursor to the next tab stop, stopping at the right margin if no further tab stops are in the line.
$L_F$	0A	Executes a line feed.
$C_R$	0D	Moves cursor to left margin of current line.
$S_O$	0E	Invokes alternate character set (line drawing).
$S_I$	0F	Invokes base character set (configured language).
X-ON( $D_1$ )	11	Causes terminal to resume transmission (see Section VI).
X-OFF( $D_3$ )	13	Causes terminal to stop transmitting all control characters except XON and XOFF.
$E_C$	1B	Introduces a control sequence.

### TERMINAL CONFIGURATION MENU OPERATIONS ("Esc&q") - saved in non-volatile memory

Esc&q <x>L      <x>  
                          0=Unlock all configuration menus  
                          1=Lock all configuration menus; plus functions <MODIFY ALL>. <BLOCK  
                          MODE>, <REMOTE MODE> and <AUTO LF> is used error message dis-  
                          played: "Function locked Press RETURN to clear").

† The following list of  $E_C$  &q escape sequences are only valid on the HP 2394A terminal.

< $E_C$  &q <x>t <y>L>      Locks or unlocks menu "x" where "x" and "y" are as follows:

#### "x"                      MENU

- 1    Datacomm
- 2    Invalid
- 3    Invalid
- 4-8   Terminal Configuration
- 9    Service Keys (valid only in lock parameter)

#### "y"                      Action

- 0                      Unlock
- 1                      Lock

† Valid on the HP 2394A terminal.

† These  $E_C$  sequences are used to change the Configuration Operations, Terminal Configuration menu, and External Device menu entry values. The active values are changed as well as those in non-volatile memory.

Escape Sequence	Menu Field	<x> value	
$E_C \& q < m > te 1 \{ < x > A$	Auto line feed	0 = OFF, 1 = ON	Modes Labels
$E_C \& q < m > te 1 \{ < x > B$	Block mode	0 = OFF, 1 = ON	
$E_C \& q < m > te 1 \{ < x > M$	Modify	0 = OFF, 1 = ON	
$E_C \& q < m > te 1 \{ < x > R$	Remote	0 = OFF, 1 = ON	
$E_C \& q < m > te 1 \{ < x > C$	Caps Lock	0 = OFF, 1 = ON	Terminal Config Menu
$E_C \& q 8 te 0 \{ < x > D$	Bell	0 = OFF, 1 = ON	
$E_C \& q < m > te 1 \{ < x > L$	Local Echo	0 = OFF, 1 = ON	
$E_C \& q < m > te 0 \{ < x > A$	XmitFnctn (A)	0 = NO, 1 = YES	
$E_C \& q < m > te 0 \{ < x > B$	SPOW (B)	0 = NO, 1 = YES	
$E_C \& q < m > te 0 \{ < x > C$	InhEolWrp (C)	0 = NO, 1 = YES	
$E_C \& q < m > te 0 \{ < x > D$	Line/Page (D)	0 = NO, 1 = PAGE	
$E_C \& q < m > te 0 \{ < x > G$	InhHndShk (G)	0 = NO, 1 = YES	
$E_C \& q < m > te 0 \{ < x > H$	InhDC2 (H)	0 = NO, 1 = YES	
$E_C \& q < m > te 0 \{ < x > I$	ASCII 8 Bits	0 = NO, 1 = YES	
$E_C \& q < m > te 0 \{ < x > N$	EscXfer (N)	0 = NO, 1 = YES	
$E_C \& q < m > te 2 \{ < x > S$	StartCol	1-80	
$E_C \& q < m > te 2 \{ < x > L$	FormsBufSize	0-255	
$E_C \& q < m > te 2 \{ < x > F$	FldSeparator	0-127	
$E_C \& q < m > te 2 \{ < x > R$	Blk Terminator	0-127	
$E_C \& q 8 te 1 \{ < x > R$	RETURN = ENTER	0 = NO, 1 = YES	
$E_C \& q 8 te 1 \{ < x > A$	RETURN Def (first character)	See note	
$E_C \& q 8 te 1 \{ < x > B$	RETURN Def (second character)	See note	
$E_C \& q 8 te 1 \{ < x > T$	Tab = Spaces	0 = NO, 1 = YES	Format Mode
$E_C \& q 8 te 1 \{ < x > U$	DATACOMM/PRINTER	0 = PORT1/PORT2 1 = PORT2/PORT1	
$E_C \& q < m > te 2 \{ < x > X$	Decimal Type	0 = US, 1 = EUR	
$E_C \& q < m > te 2 \{ < x > Y$	Implied Dec Digits	0-9	External Device
$E_C \& q < m > te 2 \{ < x > Z$	Transmit	0 = All Fields 1 = Modified fields	
$E_C \& q 8 te 1 \{ < x > N$	Printer Nulls	"x" = no. of nulls (0-255)	
Note: "x" indicates the decimal value of the ASCII code for the desired character.			
† Valid on the HP2394A Terminal			

**DATACOMM MENU ( $E_c \& q$  <TYPE> t)**

This method of configuring the datacomm portion of the terminal should be used cautiously when the escape sequence originates from a host computer. If you inadvertently misconfigure the port through which you are connected to the terminal the data link could be disabled.

The general format of the  $E_c \& q$  sequence is as follows:

```

 $E_c \& q$  <type> t
[<lock/unlock> l]
[<menu type> d]

[e]

[] {
[<BaudRate> e]
[<DataBits> h]
[<Parity> i]
[<ChkParity> j]
[<EnqAck> n]
[<SR(CH)> p]
[<Asterisk> q]
[<CS(CB)Xmit> b]
[<XmitPace> g]
[<RecvPace> h]

```

The various parameter values are as follows:

**Point-to-Point**

<BaudRate>e	0	=	110
	1	=	134.5
	2	=	150
	3	=	300
	4	=	600
	5	=	1200
	6	=	1800
	8	=	2400
	9	=	4800
	10	=	9600
	11	=	19200
<DataBits>h	0	=	7 (valid only if parity is enabled)
	1	=	8 (requires parity=none)
<PARITY>i	1	=	0's
	1	=	ODD
	2	=	1's
	3	=	EVEN
	4	=	None (valid only with 8-bit data)



**DATA COMM MENU ( $\text{E}_c\&q$  <TYPE> t) Continued**

<ChkParity>j 0 = YES  
1 = NO

<EnqAck>n 0 = NO  
1 = YES

<SR(CH)>p 0 = LO  
1 = HI

<Asterisk>q Point-to-Point  
0 = CS  
1 = DM  
2 = OFF

Internally  
0 = OFF  
1 = CS  
2 = DM

<CS(CB)Xmit>b 0 = NO  
1 = YES

<XmitPace>g 0 = None  
1 = XonXoff

<RecvPace>h 0 = None  
1 = XonXoff  
2 = TR(CD) (Not valid for full duplex modem)

## ANSI ESCAPE SEQUENCES

MNEMONIC	SEQUENCE	DESCRIPTION
<b>Cursor Control</b>		
CBT	$E_C [ Pn Z$	Cursor Back Tab
CHA	$E_C [ Pn G$	Cursor Horizontal Absolute
CNL	$E_C [ Pn E$	Cursor Next Line
CPL	$E_C [ Pn F$	Cursor Preceding Line
CPR	$E_C [ Pn; Pn R$	Cursor Position Report
CUB	$E_C [ Pn D$	Cursor Backward
CUD	$E_C [ Pn B$	Cursor Down
CUF	$E_C [ Pn C$	Cursor Forward
CUP	$E_C [ Pn; Pn H$	Cursor Position
CUU	$E_C [ Pn A$	Cursor Up
HTS	$E_C H$	Horizontal Tab Set
HVP	$E_C [ Pn; Pn f$	Horizontal and Vertical Position
IND	$E_C D$	Index
NEL	$E_C E$	Next Line
RI	$E_C M$	Reverse Index
TBC	$E_C [ Ps g$	Tab Clear
VPA	$E_C [ Pn d$	Vertical Position Absolute
VPR	$E_C [ Pn e$	Vertical Position Relative
HPA	$E_C [ Pn \backslash$	Horizontal Position Absolute
HPR	$E_C [ Pn a$	Horizontal Position Relative
HPHD	$E_C [ > 1 s$	Home Down
HPHU	$E_C [ > 0 s$	Home Up
HPSC	$E_C 7$	Save Cursor and attributes
HPRC	$E_C 8$	Restore Cursor and attributes

## ANSI ESCAPE SEQUENCES

### MNEMONIC

### SEQUENCE

### DESCRIPTION

#### Display Control

NP	$E_C [ P_n U$	Next Page
PP	$E_C [ P_n V$	Previous Page
SD	$E_C [ P_n T$	Scroll Down
SU	$E_C [ P_n S$	Scroll Up
HPSTBM	$E_C [ P_n ; P_n r$	Set top and bottom margins

#### Editing Commands

DCH	$E_C [ P_n P$	Delete Character
DL	$E_C [ P_n M$	Delete Line
ED	$E_C [ P_s J$	Erase in Display
EL	$E_C [ P_s K$	Erase in Line
IL	$E_C [ P_s L$	Insert Line

#### Character Sets

SCS	$E_C ( P_s$ or $E_C ) P_s$	Select Base Character Set Select Alternate Character Set
SGR	$E_C [ P_s ; \dots ; P_s m$	Select Graphic Rendition
HPDHL	$E_C \# 3$ (Top half) $E_C \# 4$ (Bottom half)	Make this line double-height, double-width
HPDWL	$E_C \# 6$	Make this line double-width
HPSSL	$E_C \# 5$	Make this line standard size

#### Terminal Status

DA	$E_C [ 0 c$ or $E_C [ c$	Device Attributes
DSR	$E_C [ P_s n$	Device Status Request/Report
HPID	$E_C Z$	Same as DA (above)
HPREQTPARM	$E_C [ P_s x$	Request Parameters
HPREPTPARM	$E_C [ P_s ; \dots ; P_s x$	Report Parameters

#### Terminal Control

MC	$E_C [ P_s i$	Media Copy
RIS	$E_C c$	Reset to Initial State
HPKPAM	$E_C =$	Set Keypad to Application Mode
HPKPNM	$E_C >$	Set Keypad to Numeric Mode
HPLF	$E_C [ P_s q$	Load Flags

### ANSI ESCAPE SEQUENCES

MNEMONIC	SEQUENCE	DESCRIPTION
----------	----------	-------------

#### Terminal Modes

RM	$E_C [ Ps; \dots; Ps\ 1$	Reset Mode(s)
SM	$E_C [ Ps; \dots; Ps\ h$	Set Mode(s)

Mode Name	To Set	To Reset
Cursor Keys Mode	Application $E_C [ ?\ 1\ h$	Cursor $E_C [ ?\ 1\ 1$
Keypad mode	Application $E_C =$	Numeric $E_C >$
ANSI/HP mode	HP to ANSI $\& k\ 1\ \backslash$	ANSI to HP $\& k\ 0\ \backslash$
ANSI/EM52 mode	EM52 to ANSI $E_C <$	ANSI to EM52 $E_C [ ?\ 2\ 1$
Scrolling mode	Smooth $E_C [ ?\ 4\ h$	Jump $E_C [ ?\ 4\ 1$
Origin mode	Relative $E_C [ ?\ 6\ h$	Absolute $E_C [ ?\ 6\ 1$
Wraparound	On $E_C [ ?\ 7\ h$	Off $E_C [ ?\ 7\ 1$
Keyboard Action	Kbd disabled $E_C [ 2\ h$	Kbd enabled $E_C [ 2\ 1$
Control repres.	Display fnctns $E_C [ 3\ h$	Exec fnctns $E_C [ 3\ 1$
Insert/Replace mode	Insert $E_C [ 4\ h$	Replace $E_C [ 4\ 1$
Line feed/new line	New line $E_C [ 20\ h$	Linefeed $E_C [ 20\ 1$
Multi-page Mode	Multipage $E_C [ >\ 1\ h$	Single page $E_C [ >\ 1\ 1$
Memory Lock	Locked $E_C [ >\ 2\ h$	Unlocked $E_C [ >\ 2\ 1$

### EM52 ESCAPE SEQUENCES

$E_C\ A$	Cursor Up
$E_C\ B$	Cursor Down
$E_C\ C$	Cursor Right
$E_C\ D$	Cursor Left
$E_C\ F$	Enter graphics mode
$E_C\ G$	Exit graphics mode
$E_C\ H$	Home cursor
$E_C\ I$	Reverse linefeed
$E_C\ J$	Erase to end of screen
$E_C\ K$	Erase to end of line
$E_C\ Y\ Ln\ Cn$	Direct cursor address
$E_C\ Z$	Identify
$E_C\ =$	Enter Alternate Keypad Mode
$E_C\ >$	Exit Alternate Keypad Mode
$E_C\ <$	Go to ANSI mode

# Character Sets and Keyboards

APPENDIX

B

## CHARACTER SETS

The terminal offers as standard two character sets: ROMAN 8 and Line Drawing. The ROMAN 8 set is a combination of the USASCII set and the Roman Extension set, and is shown in Table B-1 (the terminal does not display the characters located in columns 14 and 15). The Roman Extension set contains “national characters” that have a decimal value from 161 to 223. The Line Drawing Set comprises the 64 characters shown in Figure B-2.

The ROMAN 8 is always the Base Set (or Primary Set) which is invoked by default at power-on, by receipt of the Shift-in (SI) control code, or when the cursor goes into a new line (see Figure B-1).

The Line Drawing Set is always the Alternate (Secondary) Set which is invoked by receipt of a Shift-out (SO) control code (see Figure B-1).

The Shift-in character causes all subsequent characters to be displayed from the ROMAN 8 set. The terminal remains shifted in until receipt of a Shift-out control code.

The Shift-out control code causes subsequent characters to be displayed from the alternate set (Line Drawing). The terminal remains “shifted out” until receipt of a Shift-in control code or until the cursor moves to a new line. See Figure B-1.

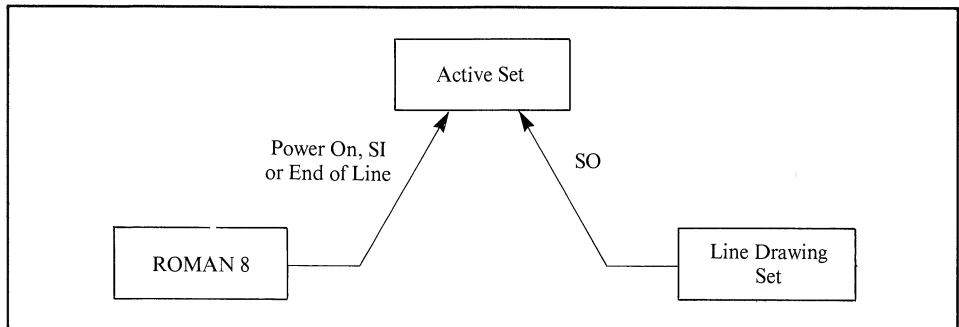


Figure B-1. Accessing ROMAN 8 and Line Drawing Set

Table B-1. ROMAN 8 Characters Supported by 2392A/2394A

					b <sub>8</sub>	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
					b <sub>7</sub>	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
					b <sub>6</sub>	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
					b <sub>5</sub>	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>			0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p				—	â	Å	Á	Đ
0	0	0	1	1	1	SOH	DC1	!	1	A	Q	a	q			À		ê	î	Ã	þ
0	0	1	0	2	2	STX	DC2	”	2	B	R	b	r			Â		ô	Ø	ã	
0	0	1	1	3	3	ETX	DC3	#	3	C	S	c	s			È	°	û	Æ	Ð	
0	1	0	0	4	4	EOT	DC4	\$	4	D	T	d	t			Ê	Ç	á	å	đ	
0	1	0	1	5	5	ENQ	NAK	%	5	E	U	e	u			Ë	ç	é	í	Í	
0	1	1	0	6	6	ACK	SYN	&	6	F	V	f	v			Î	Ñ	ó	ø	ì	–
0	1	1	1	7	7	BEL	ETB	'	7	G	W	g	w			Ï	ñ	ú	æ	Ó	$\frac{1}{4}$
1	0	0	0	8	8	BS	CAN	(	8	H	X	h	x			'	ı	à	Ä	Ò	$\frac{1}{2}$
1	0	0	1	9	9	HT	EM	)	9	I	Y	i	y			`	ı	è	ì	Õ	$\frac{3}{4}$
1	0	1	0	10	10	LF	SUB	*	:	J	Z	j	z			^	○	ò	Ö	õ	◌
1	0	1	1	11	11	VT	ESC	+	;	K	[	k	{			”	£	ù	Ü	Š	«
1	1	0	0	12	12	FF	FS	,	<	L	\	I				~	Y	ä	É	š	■
1	1	0	1	13	13	CR	GS	-	=	M	]	m	}			Û	§	ë	ï	Ú	»
1	1	1	0	14	14	SO	RS	.	>	N	^	n	~			Û	f	ö	ß	Ÿ	±
1	1	1	1	15	15	SI	US	/	?	O	_	o	DEL			£	¢	ü	Ô	ÿ	

USASCII Set

Roman Extension Set

Characters stored in  
memory but not displayed  
on screen.

Table B-2. Non-displayable Control Character Set

Character	Meaning	Decimal	Octal	Hexadecimal	Generated At Keyboard Using
NUL	Null	0	000	00	[CTRL] [ @ ]
SOH	Start Of Heading	1	001	01	[CTRL] [ A ]
STX	Start Of Text	2	002	02	[CTRL] [ B ]
ETX	End Of Text	3	003	03	[CTRL] [ C ]
EOT	End Of Transmission	4	004	04	[CTRL] [ D ]
ENQ	Enquiry	5	005	05	[CTRL] [ E ]
ACK	Acknowledge	6	006	06	[CTRL] [ F ]
BEL	Bell	7	007	07	[CTRL] [ G ]
BS	Backspace	8	010	08	[CTRL] [ H ]
HT	Horizontal Tabulation	9	011	09	[CTRL] [ I ]
LF	Line Feed	10	012	0A	[CTRL] [ J ]
VT	Vertical Tabulation	11	013	0B	[CTRL] [ K ]
FF	Form Feed	12	014	0C	[CTRL] [ L ]
CR	Carriage Return	13	015	0D	[CTRL] [ M ]
SO	Shift Out	14	016	0E	[CTRL] [ N ]
SI	Shift In	15	017	0F	[CTRL] [ O ]
DLE	Data Link Escape	16	020	10	[CTRL] [ P ]
DC1	Device Control 1	17	021	11	[CTRL] [ Q ]
DC2	Device Control 2	18	022	12	[CTRL] [ R ]
DC3	Device Control 3	19	023	13	[CTRL] [ S ]
DC4	Device Control 4	20	024	14	[CTRL] [ T ]
NAK	Negative Acknowledgement	21	025	15	[CTRL] [ U ]
SYN	Synchronous Idle	22	026	16	[CTRL] [ V ]
ETB	End Of Transmission Block	23	027	17	[CTRL] [ W ]
CAN	Cancel Line	24	030	18	[CTRL] [ X ]
EM	End Of Medium	25	031	19	[CTRL] [ Y ]
SUB	Substitute	26	032	1A	[CTRL] [ Z ]
EC	Escape	27	033	1B	[CTRL] [ [ ]
FS	File Separator	28	034	1C	[CTRL] [ \ ]
GS	Group Separator	29	035	1D	[CTRL] [ ] ]
RS	Record Separator	30	036	1E	[CTRL] [ ^ ]
US	Unit Separator	31	037	1F	[CTRL] [ _ ]
Note that RS is always displayed, whether or not display functions mode is enabled.					

Table B-3. National Characters Replaced By USASCII Characters In 7-Bit Mode

KEYBOARDS	CHARACTERS													
USASCII	#	'	<	>	@	[	\	]	^	`	{		}	~
Danish	§	'	<	>	@	Æ	Ø	Å	^	`	æ	ø	å	..
Dutch	#	'	<	>	@	ç	\	§	^	`	f		'	..
Finnish	#	'	<	>	É	Ä	Ö	Å	Ü	é	ä	ö	å	ü
Flemish and French	£	'	<	>	à	°	ç	§	^†	`	é	ù	è	..†
English Canadian and French Canadian	#	'	<	>	@	[	ç	]	^	`	é	Ç	É	..
Swiss French and Swiss German	£	'	é	è	à	°	ç	§	^	`	ä	ö	ü	..
German	£	'	<	>	§	Ä	Ö	Ü	^	`	ä	ö	ü	ß
Italian	£	'	<	>	§	°	ç	é	^	ù	à	ò	è	ì
Norwegian	#	'	<	>	@	Æ	Ø	Å	^	`	æ	ø	å	..
European Spanish	#	'†	<	>	@	í	Ñ	¿	°	'†	'	ñ	ç	..†
Latin Spanish	#	'†	<	>	@	í	Ñ	¿	^	'	'	ñ	ç	..†
Swedish	#	'	<	>	É	Ä	Ö	Å	Ü	é	ä	ö	å	ü
U. K.	£	'	<	>	@	[	\	]	^	`	{		}	~
Line Drawing Elements	T	T̄	#	≠	≡	≡	..	␣	.	≡	≡	..	␣	.
Note: In 7-bit operations, characters marked by a “†” are diacritical marks. See Table B-4 for diacritical marks in 8-bit operations.														



Table B-4. Diacritical marks In 8-bit Operation

KEYBOARDS	DIACRITIC MARKS
USASCII	None
Danish	‘            ’    “
Dutch	‘       ^       ’    “
Finnish	None
Flemish and French	^                    “
English Canadian and French Canadian	^       ’       “
Swiss French and Swiss German	^       ’       “
German	None
Italian	^
Norwegian	‘                    ’    “
European Spanish	‘                    ’    “
Latin Spanish	‘                    “
Swedish	None
U. K.	None
[Extend char] Key	‘       ^       ’    “    ~

## 7 versus 8 Bit Transmission To The Computer

On national keyboards up to 14 of the characters in the USASCII character are replaced by national characters (in order that all the characters normally associated with a particular national keyboard are available). The replaced USASCII characters used depend on the configured keyboard (as these are USASCII characters not normally employed by that keyboard). How the terminal handles the national characters for transmission and reception depends on whether it is set for 7 or 8-bit operations in the “Parity/DataBits” field in the datacomm configuration menu.

**7 bit operations.** When the terminal is configured for 7-bit operations (that is one parity bit and seven data bits), characters from the Roman Extension set cannot be directly sent to/received from a computer (as these characters are coded on eight bits, whereas USASCII characters are coded on seven bits). Consequently, the terminal automatically substitutes the corresponding USASCII character code for the Roman Extension characters in transmissions, and vice-versa for reception. (This is why the **[Extend char]** key cannot be used to access additional characters). Note that the correct Roman Extension character is displayed on the terminal.

The Roman Extension characters replacing USASCII characters depends on which national keyboard is specified in the terminal configuration menu (see table B-3).

The Line-drawing character set is affected by this as the elements correspond to USASCII characters, and some of these are not available (as they are replaced by national characters). Consequently, in 7-bit operations using national keyboards, the corresponding Roman Extension characters should be used to access the line-drawing elements (as shown in table B-3).

**8 bit operations.** When the terminal is configured for eight bit transmissions (that is no parity bit and eight data bits), all the ROMAN 8 characters can be sent to/received from the computer. The characters are coded on eight bits. Bit 8 in USASCII characters is always set to “zero”, and to “one” in Roman Extension Characters. In this case the [Extend char] key can be used to access characters that do not appear on keycaps.

When using the Line-drawing character set with national keyboards, the USASCII character that corresponds to a line element should be accessed using the [Extend char] key (see figure B-2) in order to generate the correct line-drawing element.

Line Drawing Character	!@#\$%^&*~ }{;:,.-_+`'"/\<>=
Corresponding USASCII Character	a b c d e f g h i j k l m n o p q r s t u v w x y z 0 1 2 3 4 5 6 7 8 9 - = [ \ ] ^ _ {   } ~ ; , : . - + ` ' " < > ? ! @ # \$ % & * ( ) _ { }   : ; , . - + ` ' " < > ?

With national keyboards, the USASCII characters marked “\*” (below) may not be directly available (that is, do not appear on keycaps). When the terminal is configured for 8-bit operations, these USASCII characters must be accessed using the `[Extend char]` key. When the terminal is configured for 7-bit operations, these USASCII characters are replaced by national characters that must be used instead (see Table B-3).

Figure B-2. Line Drawing Set Elements

## Data Transfers To The External Printer

In some HP printers, the Roman Extension set is accessed as an alternate (secondary) set. This secondary set can be accessed either by the shift-out control code (SO) or by the 8th bit treated as a SO control code (when the terminal is configured for 8-bit transmission). During a transfer of data to the printer, the terminal first designates the Roman Extension as the secondary set by sending the Esc)A sequence, then by sending a Shift-out (SO) control code to access the secondary set or by setting the 8th bit of the character to be sent to "1". Such printers belong to type "Extended Roman" (Printer Type = EXT ROMAN in the External Device Configuration Menu).

New HP Printers access characters from the Roman Extension directly with 8th bit = 1 (ROMAN 8 is the Base Set). Such Printers belong to type "ROMAN 8" (Printer Type = ROMAN 8 in the External Device Configuration Menu).

Table B-5. Extended Roman Data Transfers To Printer

	Printer Type = EXTROMAN	Printer Type = ROMAN 8
<b>7-bit transmission</b>	Extended Roman is secondary, accessed by <So> control code.	Not Recommended for HP Printers. (Extended Roman) characters are replaced by USASCII codes (see tables B-2 and B-3).
<b>8-bit transmission</b>	Extended Roman is secondary, accessed by 8th bit = 1.	Extended Roman is primary, accessed with full 8-bit control.

## KEYBOARDS

The terminal supports 17 different keyboards: Canadian English, Canadian French, Danish, Dutch, Finnish, French, Italian, German, Norwegian, Spanish European, Spanish Latin American, Swedish, Swiss French, Swiss German, UK, USASCII and Flemish.

The keyboard choice is specified in the "Keyboard" field in the terminal configuration menu. Figures B-4 through B-20 show the various national keyboards.

Note that the terminal is able to accommodate any keyboard layout, independently of the physical keyboard installed: the way a key is interpreted depends on the keyboard choice made in the "Keyboard" field. For example, if the "Keyboard" field is equal to "USASCII" though the keyboard layout in French, the terminal will interpret any character typed onto the keyboard as if the key in the corresponding position had been pressed on a USASCII keyboard.

Likewise, when a national keyboard has been selected, any key pressed is interpreted as the corresponding key on the appropriate national keyboard.

On a National Terminal configured for 8-bit transmission, displayed USASCII characters not available on the national keyboard and Roman Extension characters are still accessible by the use of the **[Ext Char]** key. Pressing this key gives access to a new keyboard layout where all Roman Extension characters shown in table B-1 and the 12 displaced USASCII characters are available. See figure B-3.

When a national terminal is configured for 7-bit transmission, only the characters specific to the country specified in the "Keyboard" field (see terminal configuration menu, Section II) can be accessed; in particular, displaced USASCII characters are no longer accessible.

## Diacritical Characters

Some national keyboards (e.g. French and Spanish) employ accented characters that do not appear on a keycap but which are formed by combining two characters, an accent (mute) followed by an alpha character (e.g. circumflex plus vowel, etc.).

The terminal handles these characters as follows:

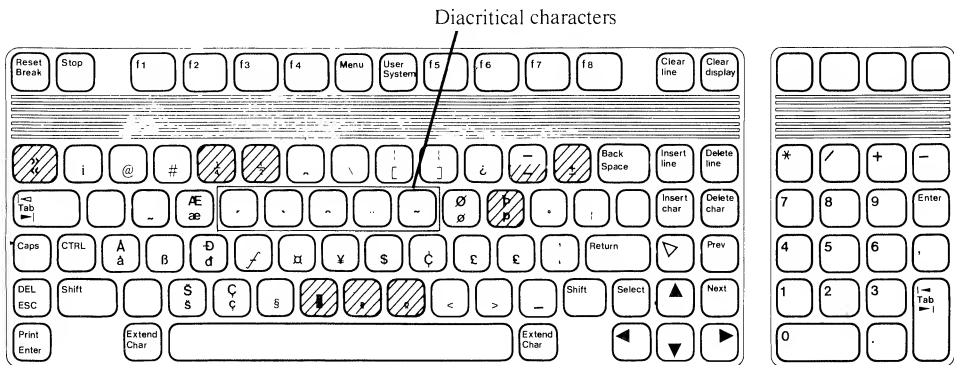
- 1) When the mute (first part of the diacritical character) is typed. For seven bit data transmission, the corresponding character code (see table B-1) is transmitted and (if echoed) the character is displayed with the cursor under it. For 8-bit transmission the character is displayed with the cursor under it but nothing is transmitted.
- 2) When the second part of the character is typed:
  - a) If this is a valid character (i.e. one giving a valid diacritical combination), the corresponding combination is displayed and the cursor moves to the next character position. In seven-bit transmission, the corresponding code of the second character is transmitted. In 8-bit transmission, the corresponding code for the combination character is transmitted.
  - b) If the space bar is pressed after the mute key, the mute character remains on the screen and the cursor moves to the next character position. In 7-bit transmission, the code for the space is transmitted. In 8-bit transmission, the code for the mute character is transmitted.
  - c) If the **[Return]** key is pressed after the mute key, in 8-bit transmission the code for the mute character is sent to the host, followed by the normal action of the **[Return]** key. In 7-bit transmission, the normal action of the **[Return]** key occurs (as the code has already been sent).
  - d) If any other key is pressed after the mute key, the mute character is erased and the normal action associated with that key is taken. For example, if it is an invalid character, it replaces the previously displayed character. In both 7-bit and 8-bit transmissions the code of this second character is transmitted.
  - e) If characters from the computer are received prior to the second keystroke, the terminal will process the received data normally. Subsequent keystrokes will be processed as if no data had been received (the mute sequence will continue). If an Escape Sequence is entered through the keyboard prior to the second keystroke, the mute character is ignored.

## LANGUAGES

The terminal supports 10 different languages: Danish, Dutch, English, Italian, French, Finnish, German, Norwegian, Spanish, Swedish. The "Language" field in the terminal configuration menu (Section II) specifies which language will be used to display the function keys labels and User Messages.

### NOTE

All languages are supported on the standard product (NO option required), this means that a French programmer may use a USASCII keyboard and still receive all the messages in French (the two fields "Keyboard" and "Language" in the terminal configuration menu (see Section II) are totally independent).



For shaded keys, the character is stored in memory but not displayed (replaced by a space).

Figure B-3. Mapping of ROMAN 8 characters (accessed by the [Extend char] key)

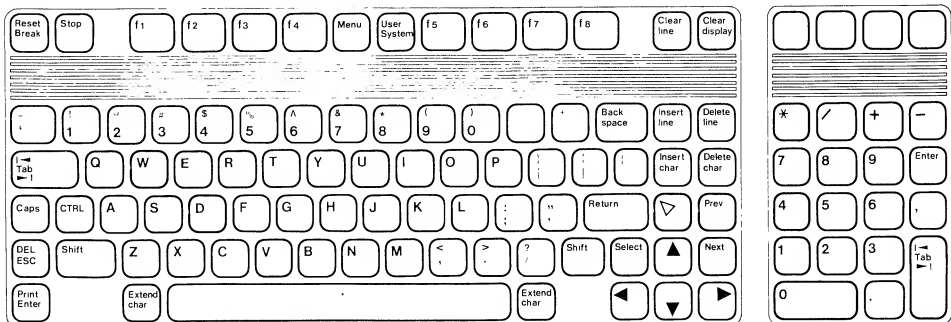


Figure B-4. American USASCII Keyboard (Standard)

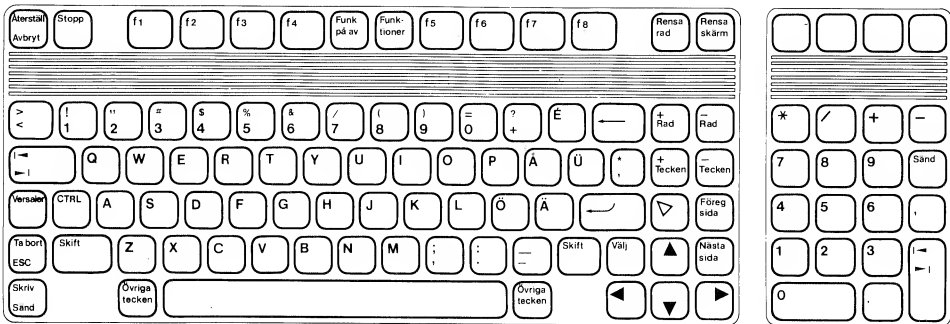


Figure B-5. Swedish Keyboard (option 101)

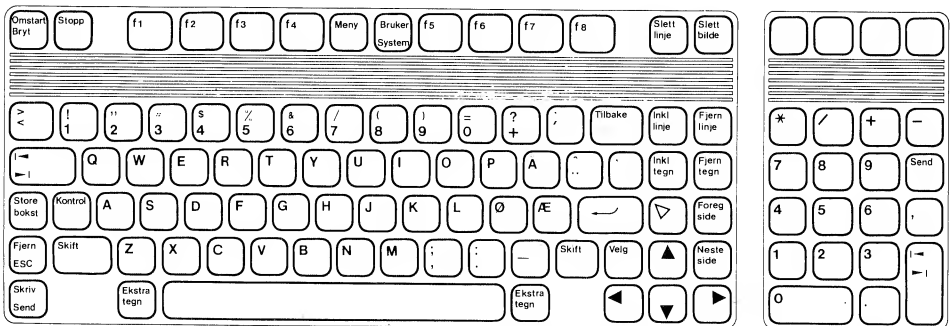


Figure B-6. Norwegian Keyboard (option 102)

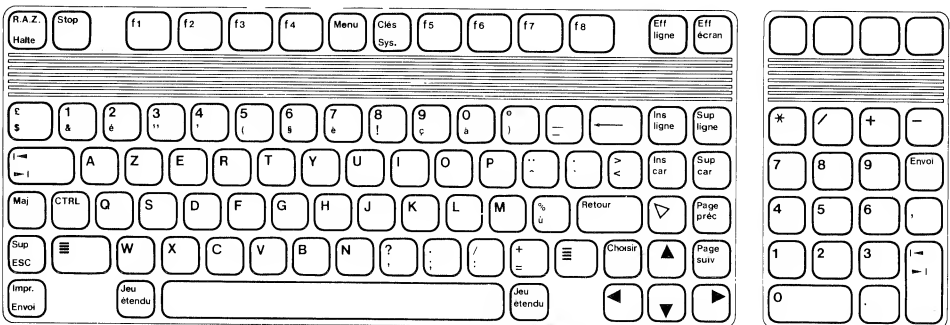


Figure B-7. French Keyboard (option 103)

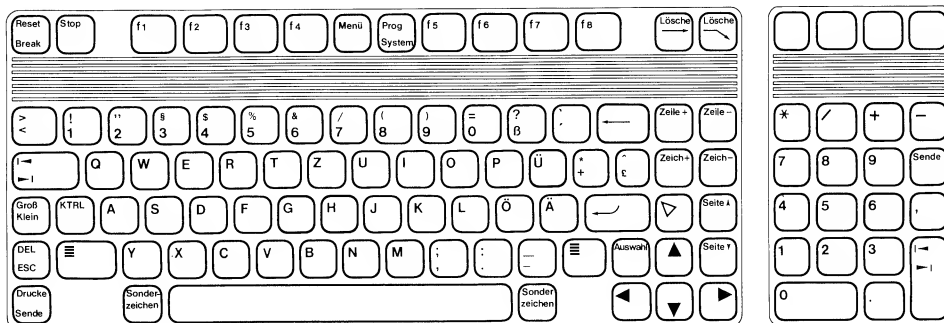


Figure B-8. German Keyboard (option 104)

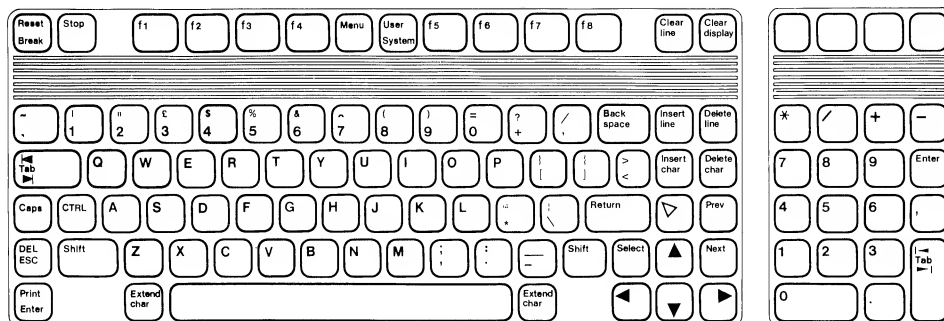


Figure B-9. English (UK) Keyboard (option 105)

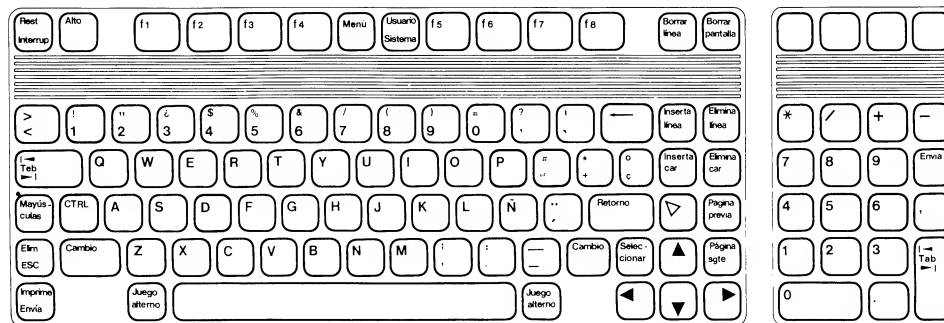


Figure B-10. Spanish (European) Keyboard (option 106)

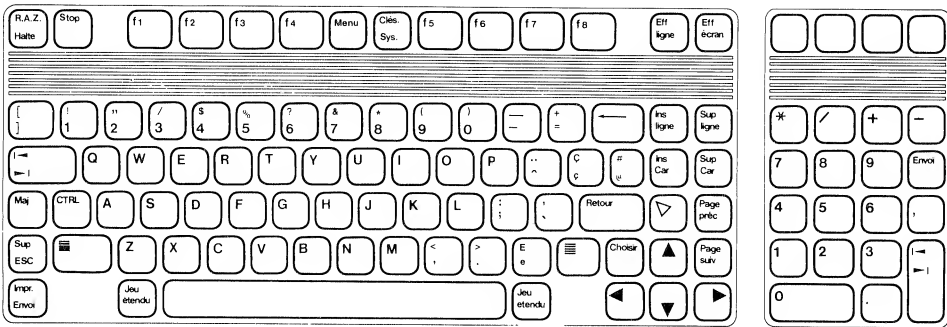


Figure B-11. Canadian (French) Keyboard (option 107)

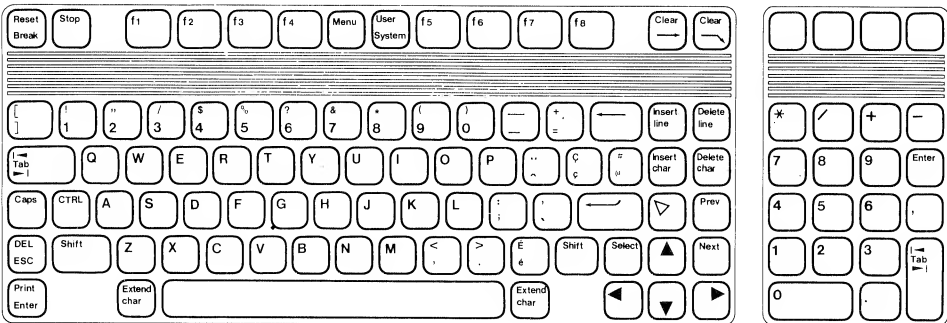


Figure B-12. Canadian (English) Keyboard (option 108)

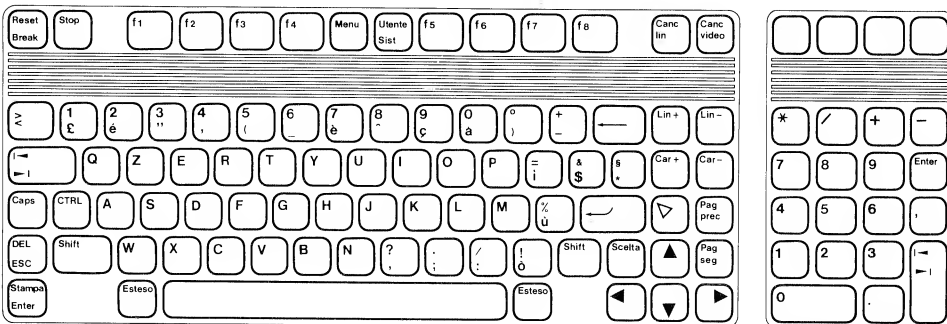


Figure B-13. Italian Keyboard (option 109)



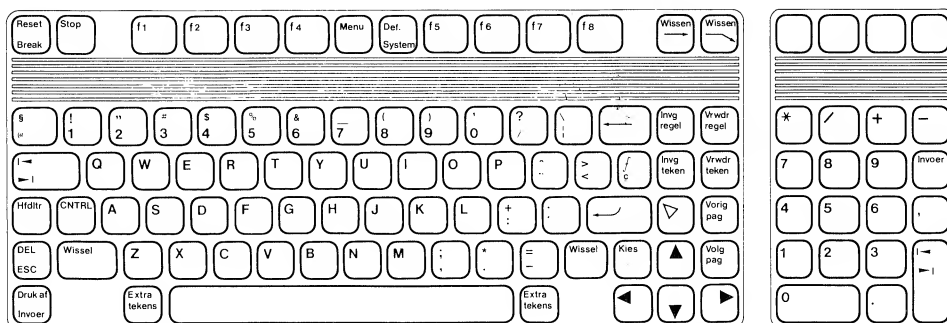


Figure B-14. Dutch Keyboard (option 110)

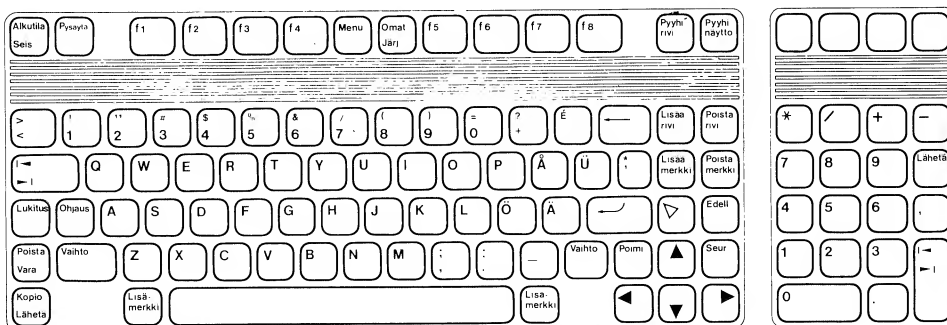


Figure B-15. Finnish Keyboard (option 111)

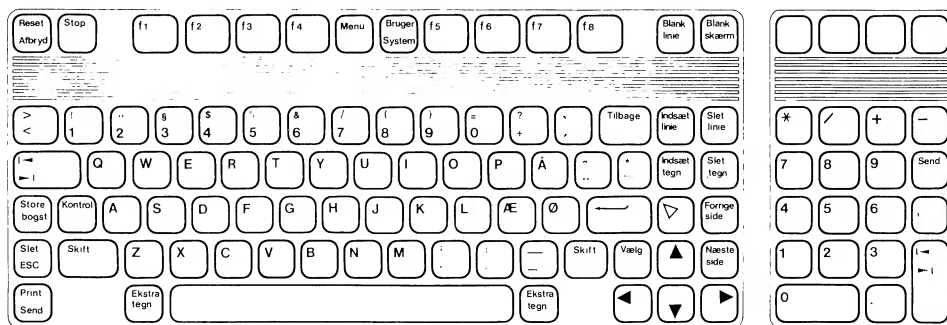


Figure B-16. Danish Keyboard (option 112)

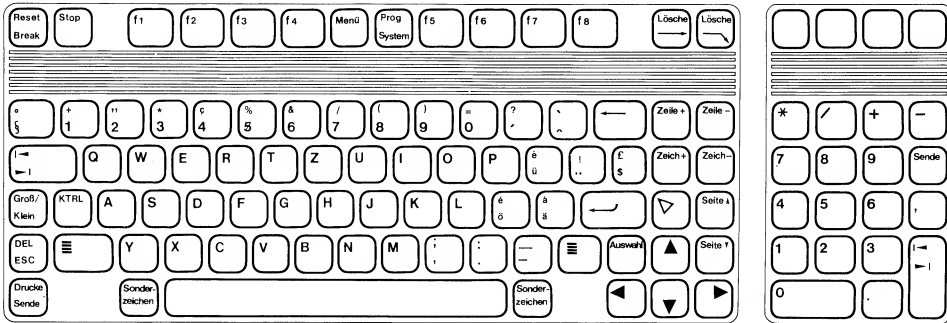


Figure B-17. Swiss (German) Keyboard (option 113)

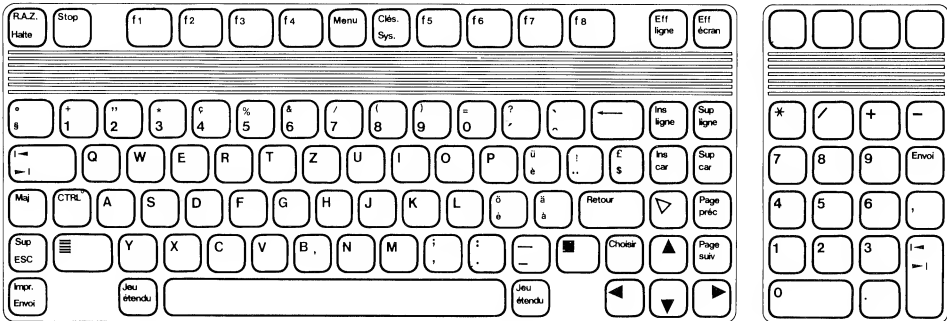


Figure B-18. Swiss (French) Keyboard (option 114)

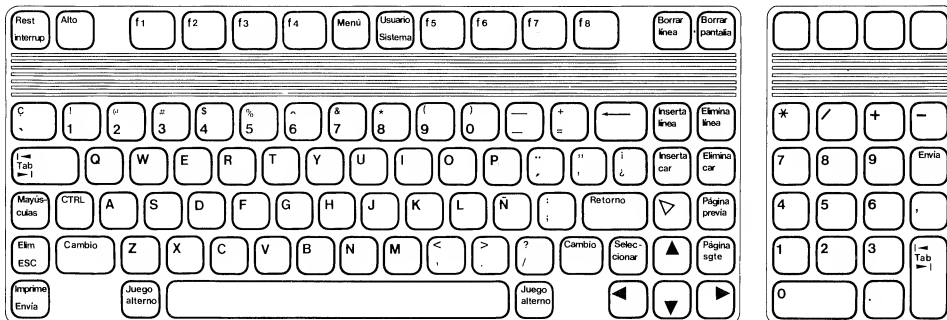


Figure B-19. Spanish (Latin American) Keyboard (option 115)

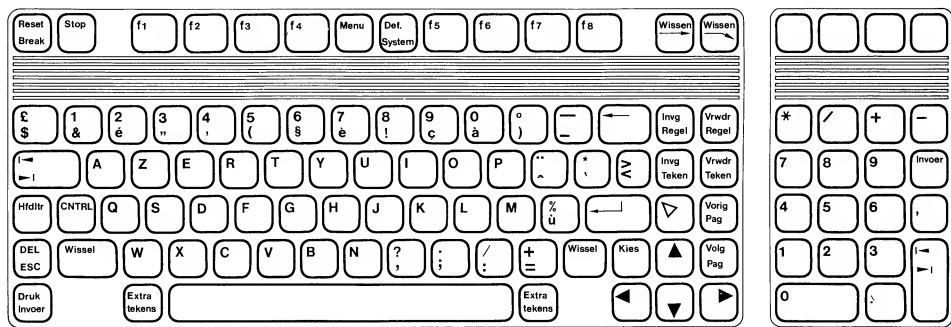


Figure B-20. Flemish Keyboard (option 116)

(

(

(

# Index

Subject	Page	Subject	Page
<b>A</b>			
absolute addressing	4-10	attribute field, user key	3-11
addressing, absolute (memory)	4-10	auto keyboard lock mode	3-11
addressing, combining absolute and relative	4-12	auto line feed mode	3-8
addressing, cursor relative	4-11		
addressing, screen relative	4-10	<b>B</b>	
advance line	5-4	backspace definition field	10-13
advance page	5-4	backtab	4-20
alphanumeric capabilities status	7-12, 7-13	battery maintenance	9-1
alternate character set, extended character	B-6	baud rate, field, datacomm	
alternate character set, line drawing	4-31, B-1	configuration menu	6-8
alternate character set, line drawing with function keys	4-34, 4-35	baud rate, field, external device	
alternate character sets, ANSI mode	10-10	configuration menu	5-11, 5-12
alternate character sets, EM52 mode	10-10	bell, code	3-25
alternate character sets, HP mode	10-9	bell, terminal configuration menu field	2-2, 2-4
ANSI character set selection sequences	10-27	binary transfers	5-7
ANSI Configuration menu	10-11	blinking video	4-27
ANSI control sequences	10-15	block mode	3-3
ANSI cursor control sequences	10-18	block terminator field, terminal	
ANSI display control sequences	10-24	configuration menu	2-2, 2-3
ANSI editing control sequences	10-26	bottom logging	5-3
ANSI escape sequences	10-41	break	3-24
ANSI mode	10-4	brightness control	1-4
ANSI mode alternate character sets	10-10	buffer, overflow	6-12
ANSI mode escape sequences	10-5	buffer, receive	6-12
ANSI terminal control sequences	10-31		
ANSI terminal mode selection sequences	10-33	<b>C</b>	
ANSI terminal status sequences	10-29	cables, data communications	6-3
ANSI/EM52/HP mode configuration	10-1	cables, external device	5-9
Answerback Message field	10-14	caps mode ([Caps]) key)	3-9
ASCII character set	B-2	CBT	10-18
ASCII transfers	5-7	CHA	10-19
asterisk, field, datacomm		character mode, line modify mode	3-1
configuration menu	6-8, 6-9	character mode, modify all mode	3-2
asterisk, screen labels	1-12	character mode, selection	3-3
asterisk, screen status line	1-14	character set selection sequences, ANSI	10-27
asterisk, terminal configuration		character set, alternate line drawing	4-31, B-1
menu field	2-3, 2-4	character set, available	B-1
		character set, extended characters	B-6

# Index continued

Subject	Page
characters, control	10-16
check parity field, datacomm	
configuration menu	6-8, 6-9
checking parity	6-11
cleaning the terminal	9-1
clear display	4-17
clear line	4-18
clearing margins	4-18
clearing tabs	4-19
CNL	10-19
combining absolute and relative addressing	4-12
completion codes, device control	5-6
computer port	1-16
config keys label	2-1
configuration menus, datacomm	6-8
configuration menus, external device	5-11
configuration menus, lock/unlock	2-13
configuration menus, programming	2-14, A-4
configuration menus, terminal	2-2, 2-3
configuration menus, user-keys definition	3-12
configuring the terminal	1-14, 2-2, 2-3, 5-8
control characters	A-13
control characters	10-16
control sequence summary, EM52	10-38
control sequences, ANSI	10-15
copy all	5-5
copy line	5-5
copy memory to device	5-6
copy menu	5-6
copy page	5-5
CPL	10-19
CPR	10-19
CRM	10-34
CS (CB) Xmit field, datacomm	
configuration menu	6-8, 6-10
CS (CB) Xmit field, external device	
configuration menu	5-11, 5-13
CUB	10-19
CUD	10-19
CUF	10-20
CUP	10-20
cursor control	4-2
cursor control keys	10-7
cursor control operation	A-4

Subject	Page
cursor control sequences, ANSI	10-18
cursor down	4-3
cursor home down	4-3
cursor home up	4-2
cursor left	4-4
cursor position mode	3-10
cursor relative addressing	4-11
cursor right	4-4
cursor sensing	4-9
cursor up	4-3
CUU	10-20

## D

DA	10-29
data bits, field, datacomm	
configuration menu	6-8, 6-9
data bits, field, external device	
configuration menu	5-11, 5-12
data bits, start/stop	6-11
data communications, cables	6-3
data communications, configuration menu	6-8
data communications, description	6-2
data communications, test hood	8-6
data logging, bottom	5-3
data logging, top	5-3
data transfers, computer to device	5-7
data transfers, display to device	5-4
datacomm module	1-7
datacomm/ext dev field, terminal	
configuration menu	2-2, 2-3, 2-4
DC1/DC2 handshake	2-8, 3-17, 5-13, 6
DCH	10-26
decimal type	2-7
defining fields from the keyboard	4-43
defining fields programmatically	4-46
defining user keys locally	3-12
defining user keys programmatically	3-13
delete character	4-16
delete line	4-14
designing forms	4-30, 4-34
device configuration menu	5-11, 5-12
device control completion codes	5-6
device control labels	1-12, 5-4
device modes labels	1-11, 5-2
device port	1-16

## Index continued

Subject	Page
device status .....	7-17
diacritic characters .....	B-8
disable display .....	4-27, A-10, A-11
disable keyboard .....	3-11, 3-16, A-2
display control sequences, ANSI .....	10-24
display enhancements .....	4-26
display enhancements using the system	
function keys .....	4-29
display functions mode .....	3-7
display lock .....	3-6, A-11
DL .....	10-26
drawing forms .....	4-31
drawing lines .....	4-35
DSR .....	10-30

### E

echo .....	2-7
ED .....	10-27
edit operations .....	4-13, 4-40
editing control sequences, ANSI .....	10-26
EL .....	10-27
EM52 control sequence summary .....	10-38
EM52 escape sequences .....	10-43
EM52 mode .....	10-3
EM52 mode alternate character sets .....	10-10
EM52 mode escape sequences .....	10-5
enable keyboard .....	3-11, 3-15, A-2
enhancements display .....	4-26
EnqAck field, datacomm	
configuration menu .....	6-8, 6-9
EnqAck handshake .....	5-7, 6-14
error messages .....	8-1
errors, receive .....	6-12
escape sequences, ANSI .....	10-5
escape sequences, ANSI .....	10-41
escape sequences, available .....	A-1
escape sequences, EM52 .....	10-43
escape sequences, EM52 mode .....	10-5
escape sequences, from the keyboard .....	10-6
escape sequences, HP mode .....	10-5
escape sequences, transfer to computer .....	2-8
escape sequences, transfer to printer .....	2-12

Subject	Page
EscXfer (N) field, terminal	
configuration menu .....	2-2, 2-3, 2-12
Extend char key .....	3-24
external device, cabling .....	5-9
external device, configuration menu ....	5-11, 5-12
external device, control labels .....	5-4
external device, port .....	1-16
external device, status .....	7-17

### F

feature set .....	1-1, 1-2
field attribute .....	4-40
field definition menu .....	4-44
field, Answerback Message .....	10-14
field, Backspace Definition .....	10-13
field, Multipage .....	10-13
fields, protected and unprotected .....	4-33, 4-40
FldSeparator field, terminal	
configuration menu .....	2-2, 2-3, 2-11
format mode tab sequence control .....	4-33
forms (format) mode .....	4-33
forms buf size .....	2-6
forms cache, directory .....	4-52, 4-53
forms cache, status .....	7-19
forms, designing and using .....	4-30, 4-39
dull duplex operations .....	6-13
function key hierarchy .....	1-12, 1-13
function keys .....	1-9

### H

half bright .....	4-26, 4-29
handshaking .....	2-8, 6-15
hard reset .....	3-23
hierarchy, function keys .....	1-12, 1-13
home down .....	4-3
home up .....	4-2
HP mode .....	10-3
HP mode alternate character sets .....	10-9
HP mode escape sequences .....	10-5
HPA .....	10-22
HPANM .....	10-36
HPAWM .....	10-37

## Index Continued

Subject	Page
HPCKM .....	10-35
HPDHL .....	10-28
HPDWL .....	10-29
HPHD .....	10-23
HPHU .....	10-23
HPID .....	10-30
HPKPAM .....	10-32
HPKPNM .....	10-32
HPLF .....	10-32
HPMLM .....	10-37
HPMPM .....	10-37
HPOM .....	10-36
HPR .....	10-23
HPRC .....	10-23
HPREQTPARM .....	10-30
HPRPTPARM .....	10-31
HPSC .....	10-23
HPSCLM .....	10-36
HPSSL .....	10-29
HPSTBM .....	10-26
HTS .....	10-20
HVP .....	10-20

### I

ID status .....	7-2
identify ROMs .....	8-5
IL .....	10-27
implied decimals .....	2-6, 4-29
IND .....	10-21
InhDC2 (H) field, terminal configuration menu .....	2-2, 2-3, 2-8
InhEolWrp (C) field, terminal configuration menu .....	2-2, 2-3, 2-6
InhHndShk (G) field, terminal configuration menu .....	2-2, 2-3, 2-8
insert character .....	4-14
insert line .....	4-13
installed memory status .....	7-12, 7-15
installing a point-to-point configuration .....	6-7
interface capabilities status .....	7-12, 7-16
inverse video .....	4-26, 4-29
IRM .....	10-35

Subject	Page
---------	------

### K

KAM .....	10-34
key definition field, user key .....	3-11
keyboard, auto-lock mode .....	3-11
keyboard, cleaning .....	9-1
keyboard, control .....	3-1
keyboard, enabling/disabling .....	3-11, 3-15, A-2
keyboard, field, terminal configuration menu .....	2-2, 2-3
keyboard, groups of keys .....	1-8
keyboard, lock .....	3-11, 3-15, A-2
keyboard, national .....	B-7
keypad, numeric .....	10-8
keys, cursor control .....	10-7
keys, functions keys .....	1-9
keys, [Extend char] .....	3-24
keys, [Menu] .....	1-14, 3-1, 3-12
keys, [Print Enter] .....	3-16
keys, [Select] .....	3-24
keys, [Stop] .....	3-25
keys, [User System] .....	1-9, 3-1

### L

label field, user key .....	3-12
labels, hierarchy function .....	1-12, 1-13
language field, terminal configuration menu .....	2-2, 2-3
line modify mode .....	3-1
line types .....	4-34
Line/Page (D) field, terminal configuration menu .....	2-2, 2-3, 2-11
lines, drawing .....	4-35
LNM .....	10-35
Local echo field, terminal configuration menu .....	2-2, 2-3, 2-7
local mode .....	3-5, 6-13
lock, configuration .....	2-13
lock, display .....	3-6, A-12
lock, keyboard .....	1-14, 3-11, 3-15
logging, bottom .....	5-3
logging, top .....	5-3



## Index continued

Subject	Page	Subject	Page
<b>M</b>			
margins, setting	4-35	modify mode, line	3-1
margins, setting and cleaning	4-17	Multipage field	10-13
MC	10-31	<b>N</b>	
memory	1-5, 1-6	national keyboards	B-7
memory addressing	4-6	NEL	10-21
memory lock mode	3-6	next page	4-5
Menu key	1-12, 1-13, 3-12	NP	10-24
menu, copying to a printer	5-6	numeric keypad	10-8
menu, datacomm configuration	6-8	<b>O</b>	
menu, external device configuration	5-11, 5-12	options	1-6
menu, terminal	2-2, 2-3	overflow protect	3-6
menu, user key definition	3-13	<b>P</b>	
messages, terminal error	8-1	pacing mechanism	6-14
mode, ANSI	10-4	page, next/previous	4-5
mode, auto keyboard lock	3-11	parity checking	6-11
mode, auto line feed	3-8	Parity/Databits field, datacomm	
mode, block	3-3	configuration menu	6-8, 6-9
mode, caps	3-9	Parity/Databits field, external	
mode, caps lock	3-9	device configuration menu	5-11, 5-12
mode, character	3-3	point-to-point configuration installation	6-7
mode, display functions	3-7	point-to-point programming information	6-11
mode, display lock/overflow protect	3-6	port test	8-5
mode, EM52	10-3	power-on test	8-2
mode, forms (format)	3-10	PP	10-24
mode, HP	10-3	previous page	4-5
mode, line modify	3-1	primary terminal status	7-3
mode, local	3-5	print	2-7
mode, logging (bottom/top)	5-3	Print/Enter key	3-16
mode, memory lock	3-6	printer control	5-1
mode, modify all	3-2	Printer Nulls field, external	
mode, record	5-1	device configuration menu	5-11, 5-12
mode, remote	3-5	Printer Type field, external	
mode, send cursor position	3-10	device configuration menu	5-11, 5-12, B-7
mode, smooth scroll	3-5	printer, cabling	5-9
modem	6-5	printer, copying to	5-5
modem disconnect	3-25	printer, data transfers to	5-4, A-6
modem indicator	1-14, 6-9	printer, port	1-16
modes label	1-11		
modified fields, transmit	4-48		
modify all mode	3-2		

## Index continued

Subject	Page	Subject	Page
printer, receiving data from computer	5-7	setting margins	4-18, 4-35
printer, selecting as destination	5-4	setting modified data tags	4-48
printer, transferring escape sequence	2-12	setting tabs	4-19
priority, data transfers	6-16	setting tabs	10-14
protected fields	4-33	SGR	10-28
<b>R</b>		shift-in code	B-1
receive buffer	6-12	shift-out code	B-1
receive errors	6-12	SM	10-34
record mode	5-1	smooth scroll mode	3-5
RecvPace field, datacomm		soft reset	3-23
configuration menu	6-8, 6-10	specifications	1-5
relative addressing	4-9	SPOW (B) field, terminal	
relative addressing, cursor	4-11	configuration menu	2-2, 2-3, 2-5, 2-1
relative addressing, screen	4-10	SR (CH) field, datacomm	
remote mode	3-5, 6-13	configuration menu	6-8, 6-9
reset, hard	3-23	SRR Invert field, external	
reset, soft	3-23	device configuration menu	5-11, 5-13
Reset/Break key	3-23	SRRXmit field, external	
Return Default field, terminal		device configuration menu	5-11, 5-13
configuration menu	2-2, 2-12	Start Col field, terminal	
Return = Enter	2-6	configuration menu	2-2, 2-3, 2-8
RI	10-21	status, alphanumeric capabilities	7-12, 7-13
RIS	10-31	status, device	7-17
RM	10-34	status, forms cache	7-19
roll text, down	4-5	status, graphics capabilities	7-12, 7-14
roll text, up	4-4	status, ID	7-2
ROMs, identify	8-5	status, Installed memory	7-12, 7-15
<b>S</b>		status, Interface capabilities	7-12, 7-16
saving tabs	10-14	status, primary	7-3
screen, cleaning	9-1	status, secondary	7-3
screen, labels	1-9, 1-10, 1-13	stop key	3-25
scrolling	4-4, 4-5	stop/start bits	6-11
SCS	10-28	SU	10-25
SD	10-25	<b>T</b>	
secondary status	7-3	tab, back	4-20
Select key	3-24	tab, forward	4-20
selecting the terminal mode		tab, sequence control	4-21
programmatically	10-6	tab, setting/clearing	4-19
self tests, identify ROMs	8-5	tab, tab = spaces	2-6
self tests, ports	8-6, 8-7	Tab = Spaces	2-6
self tests, power-on	8-2	tabs, saving	10-14
send cursor position mode	3-10	tabs, setting	10-14
send display	3-17	TBC	10-22
		terminal configuration menu field	2-5, 3-9

**Index** continued

<b>U</b>	
underline enhancement . . . . .	4-26
unlock configuration . . . . .	2-13
unprotected fields . . . . .	4-33, 4-40
user key definition menu . . . . .	3-13

Subject	Page
user keys, defining locally . . . . .	3-12
user keys, defining programmatically . . . . .	3-13
user-definable keys . . . . .	3-11
User/System key . . . . .	1-9

video enhancements . . . . .	4-26
VPA . . . . .	10-22
VPR . . . . .	10-22

## wait code ..... 3-25

X-ON/X-OFF handshaking .....	6-14
Xmit Pace field external device configuration menu .....	5-11, 5-13
Xmit Pace field, datacomm configuration menu .....	6-8, 6-10
XmitFnctn (A) field, terminal configuration menu .....	2-2, 2-8

( )

( )

( )